

PHP典型模块精解

快速服务：微博、QQ在线服务

自学视频：72集大型多媒体自学视频

海量资源：模块库、案例库、素材库、题库

潘凯华　李　慧　刘　欣　等编著

本书提供了内容丰富的配套资源，可以登录www.tup.com.cn，找到本书后，在该页面的“网络资源”超链接处下载。也可以访问本书的新浪微博，根据提示链接下载。

清华大学出版社

PHP 典型模块精解

潘凯华 李 慧 刘 欣 等编著

（72 集大型多媒体自学视频）

（模块库、案例库、素材库、题库）

（微博、QQ、论坛技术支持）

清华大学出版社

北 京

内 容 简 介

本书以注册登录、分页显示、上传与下载、网页计数器、购物车、在线支付、假日公告管理、投票系统、留言本、在线论坛、博客、在线编程词典、邮件发送系统、FTP 管理系统和创建 PDF 文档 15 个模块为案例，从 Web 项目开发的角度出发，打造点、线、面一体化的学习架构，让读者既能够学习到子功能模块的实现方法、开发技巧，同时又能够合理地运用子功能模块，打造出完整的项目。

本书模块涉及面广，实用性强。通过对本书的学习，读者可以了解 Web 开发中各个环节的特点、功能，以及如何合理地连接各个环节，打造出完整的项目。另外，本书中所提供的模块不仅包括常用功能模块，还包含目前比较流行的热点模块，既可达到读者二次开发的目的，同时亦可满足读者对流行、热点内容的好奇心。

本书提供了大量的自学视频、源程序、素材，提供了相关的模块库、案例库、素材库、题库等多种形式的辅助学习资料，还提供迅速及时的微博、QQ、论坛等技术支持。

本书模块丰富，讲解细致，非常适合作为零基础学习人员的学习用书和大中专院校师生的学习教材，也适合作为相关培训机构的师生和软件开发人员的参考资料。

图书在版编目（CIP）数据

PHP 典型模块精解/潘凯华，李慧，刘欣等编著．—北京：清华大学出版社，2012.2

ISBN 978-7-302-27205-2

I. ①P…　II. ①潘…　②李…　③刘…　III. ①PHP 语言–程序设计　IV. ①TP312

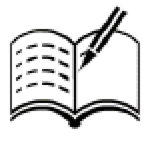
中国版本图书馆 CIP 数据核字（2011）第 223021 号

责任编辑：赵洛育　刘利民
版式设计：文森时代
责任校对：姜　彦
责任印制：
出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
　　社 总 机：010-62770175　　**邮　　购：**010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　**印　　张：**25.75　**字　　数：**595 千字
版　　次：2012 年 2 月第 1 版　**印　　次：**2012 年 2 月第 1 次印刷
印　　数：1～5000
定　　价：49.80 元

产品编号：043924-01

前　言

Preface



众里寻她千百度，蓦然回首，那人却在灯火阑珊处！

每个想要成为程序员的、已经是初级程序员的，或者已经是中级程序员的读者，都有这样一个梦想，期待着自己有一天能够成为一名高级程序设计大师。

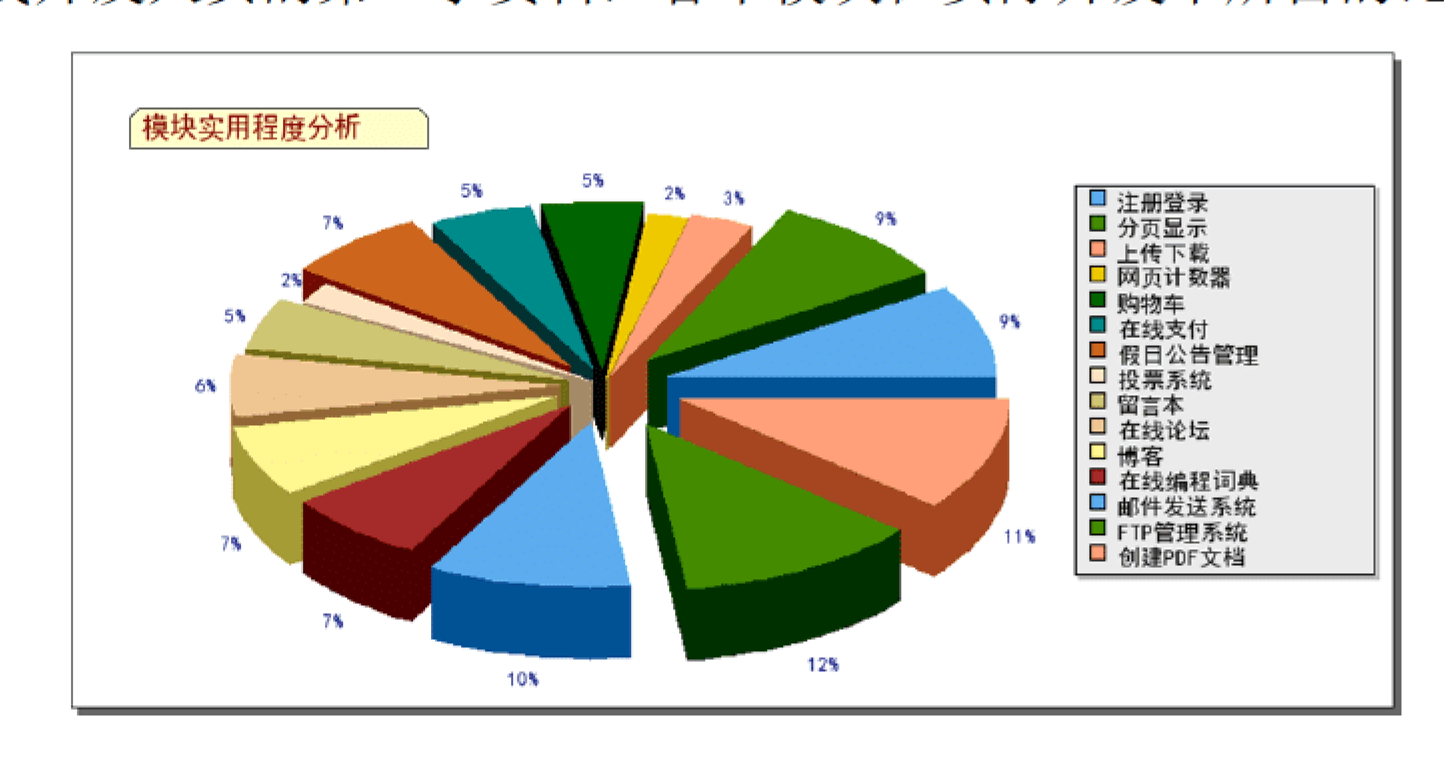
每个人都在为这个理想而奋斗着，不断地付出，开发着各种各样的项目、模块，以此来积累经验，提高技术水平。当有一天梦想实现的时候，回想起自己是如何做到的，蓦然发现正是那些各种各样的项目、模块成就了自己。

正所谓，众里寻她千百度，蓦然回首，那人却在灯火阑珊处！

本书精选了 15 个模块，直接将这灯火阑珊处的“她”呈现给诸位读者，希望读者能够早日成就梦想。

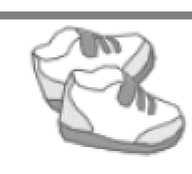
本书内容

本书分为 15 章，包含 80 项开发技术、经验，涵盖注册登录、分页显示、上传与下载、网页计数器、购物车、在线支付、假日公告管理、投票系统、留言本、在线论坛、博客、在线编程词典、邮件发送系统、FTP 管理系统和创建 PDF 文档等多个模块的内容。所选模块均来源于一线开发人员的第一手资料，各个模块在实际开发中所占的比重如下图所示。



本书特色

- ☑ 纵向延伸。本书精选了 15 个模块，涵盖项目开发中应用的各方面技术和数据库，体现了从小到大、从易到难的学习过程。
- ☑ 横向连接。属于 PHP 系列图书的一部分，是 PHP 入门学习、范例练习的延伸，以





模块开发的形式锻炼读者 Web 开发的能力，并且逐步向更高级别的项目开发迈进。

☑ 易学、易用。本书中的每个模块都通过“技术提炼”栏目，对代码中重要的对象、方法、语句和重点知识等进行说明，以便于读者在阅读代码时透彻理解代码的含义和相关技术、技巧。

☑ 赠送所有模块源代码。书中所有模块均提供源代码，用户在开发中可以快速借鉴或应用。

☑ 提供 72 集（节）视频录像。本书提供了 72 集（节）视频录像，读者可以边看边学，快速提高。

本书配套资源

本书提供了内容丰富的配套资源，包括自学视频、源程序、素材，以及模块库、案例库、题库、素材库等多项辅助内容，读者朋友可以通过如下方式获取。

第 1 种方式：

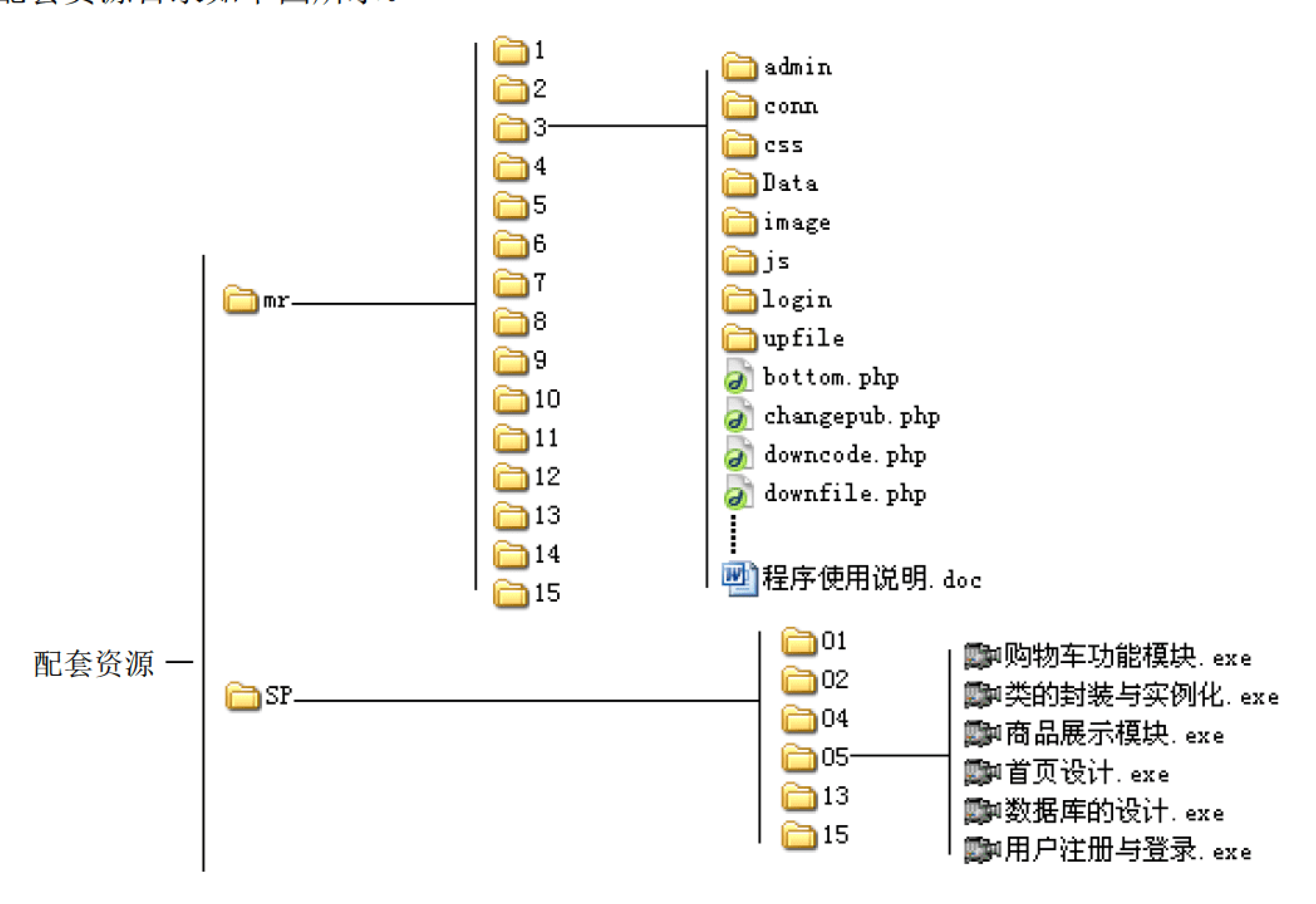
（1）登录 www.tup.com.cn，在网页右上角的搜索文本框中输入本书书名（注意区分大小写和留出空格），或者输入本书关键字，或者输入本书 ISBN 号（注意去掉 ISBN 号间隔线“-”），单击“搜索”按钮。

（2）找到本书后单击超链接，在该书的网页下侧单击“网络资源”超链接，即可下载。

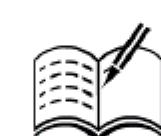
第 2 种方式：

访问本书的新浪微博 PHPbook，找到配套资源的链接地址进行下载。

配套资源目录如下图所示。







本书约定

☑ **项目使用方法**

在学习本书的过程中，可以从配套资源中复制模块，去掉其只读属性。有些模块需要使用相应的数据库或第三方资源，此类模块在使用前需要进行相应配置，详细使用方式参见配套资源中的使用说明。此外，如果直接将本书内容用于商业用途，由此产生的不良后果由用户自己承担。

☑ **部分项目只给出关键代码**

由于篇幅限制，书中有些模块只给出了关键代码，完整代码参见配套资源中的对应程序。

读者人群

本书非常适合以下人员阅读：

☑ 从事 PHP 编程行业的开发人员

☑ 有一定语言基础，想进一步提高技能的人员

☑ 大中专院校的老师和学生

☑ 即将走上工作岗位的大学毕业生

☑ 相关培训机构的老师和学员

☑ PHP 编程爱好者

读者服务&本书勘误

读者在使用本书过程中遇到的所有问题，均可通过以下方式联系我们。

1．新浪微博：PHPbook。

及时发布读者答疑、本书勘误、配套资料更新等内容。

2．腾讯 QQ：4006751066。

3．登录网站：www.mingribook.com，在论坛、勘误发布、读者纠错、技术支持、读者之家等栏目中的相关模块中提问、留言或查看。

本书作者

本书由明日科技组织编写，参与编写的有潘凯华、李慧、刘欣、高春艳、王小科、王军、赵会东、董大永、尹强、李继业、张磊、赛奎春、宋坤、刘锐宁、梁水、李伟明、刘彬彬、安剑、孙秀梅、巩建华、房大伟、吕双、刘云峰、王国辉、李钟尉、陈丹丹、卢翰、梁晓岚、杨丽、顾丽丽、刘龄龄、王乐乐、陈英、李贺、朱晓、肖鑫、李鑫和刘冠男等。由于作者水平有限，疏漏和不足之处在所难免，敬请广大读者朋友批评指正。

编　者

目　录

Contents






Note




Note




Note




Note

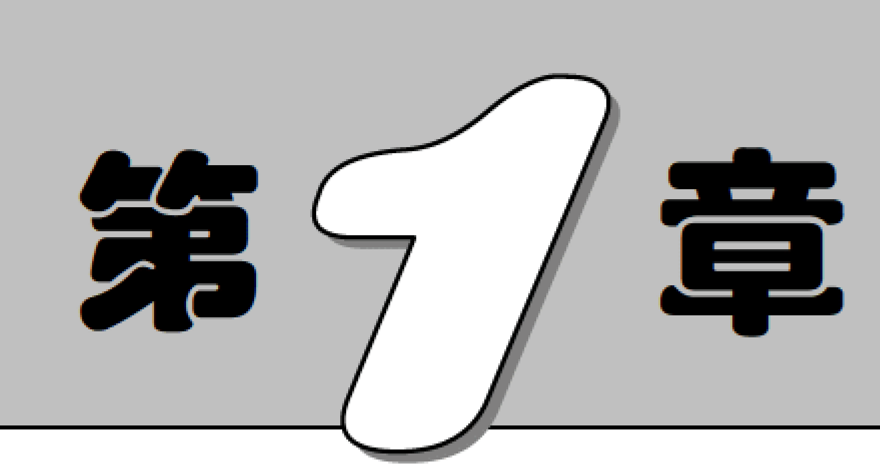



Note





Note

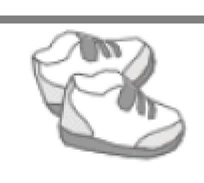


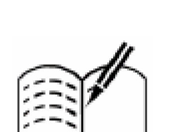
注册登录模块

（自学视频、源程序：配套资源\mr\1\）

从动态网站诞生之初，注册登录模块就成为网站必不可少的组成部分，并且以其功能单一、实现相对简单的特点，成为入门级的模块。随着 Web 2.0 的发展，人机交互成为主流趋势，注册登录模块也顺应了时代的发展，变得更加有“亲和力”，用户那种花了许久时间填写注册表单，提交后却发现信息填写错误，需要重新填写的惨痛经历越来越少，随着 Ajax 技术的兴起，这种情况就更少发生了。本章将开发一个功能齐全并有着良好交互性的注册登录模块。通过本章的学习，读者能够学到以下内容。

- 注册信息实时提示
- E-mail 激活技术
- GD2 验证码
- 更换验证码
- 防 SQL 注入技术





Note

1.1 注册及登录验证概述

注册及登录验证模块（以下简称注册登录模块）的功能比较单一，除了用于验证用户信息的有效性和一些必要的安全设置外，更多的是解决如何提高程序的可操作性，以更好地方便用户。注册登录模块主要由用户注册、用户登录和找回密码 3 部分组成。

1.1.1 用户注册流程

用户注册的信息分为必填信息和可选信息。必填信息默认是可见的，而可选信息默认是隐藏的。在必填信息不完整的情况下，"注册"按钮是不可用的。对于可选信息，用户可以选择填写或不填写。但如果填写了，那么就需要对信息的正确性进行检验。当注册成功后，系统将向用户所填写的 E-mail 发送一封激活邮件，用户只有激活后，才可使用该账号进行登录。用户注册流程图如图 1.1 所示。

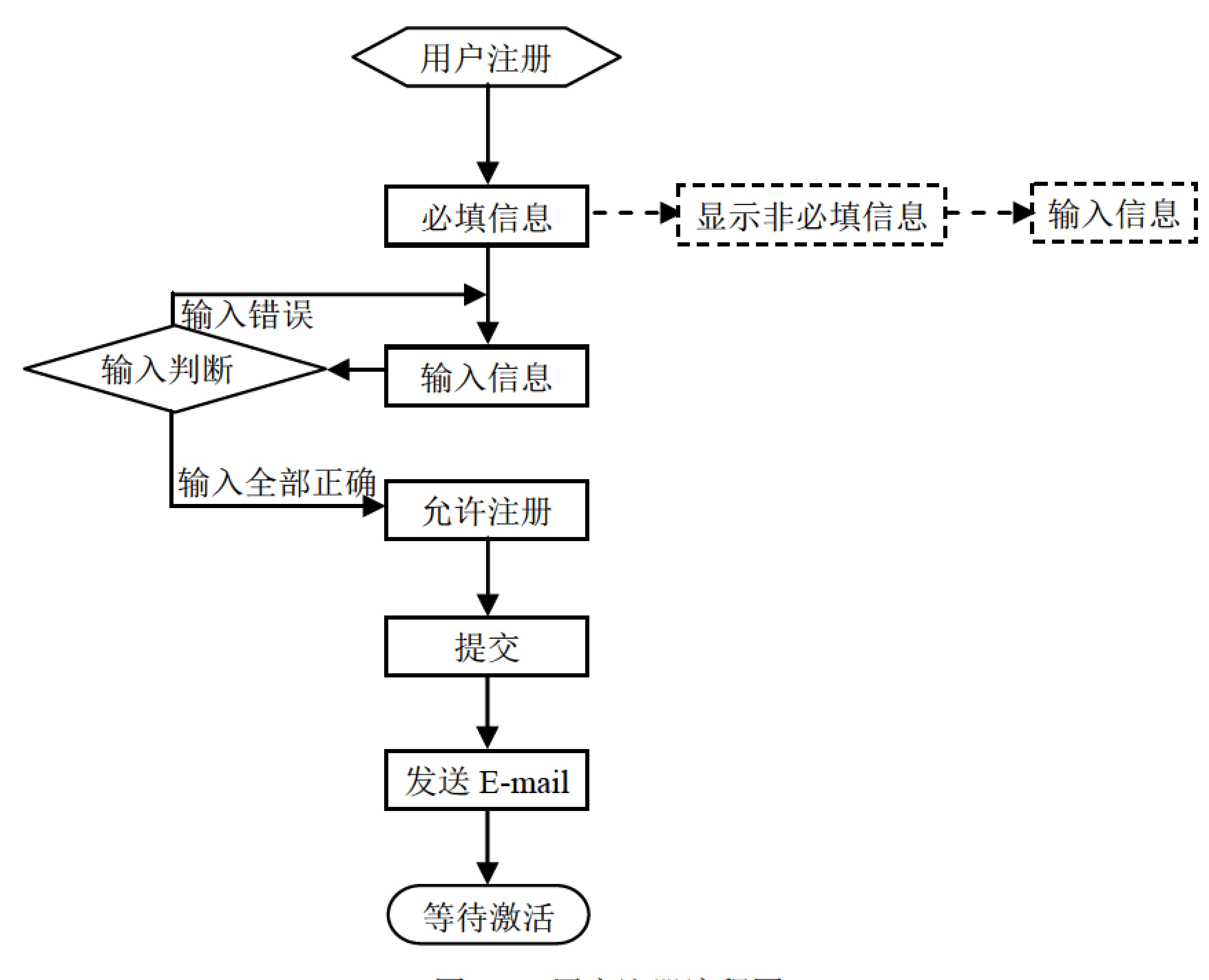


图 1.1　用户注册流程图

1.1.2 用户登录流程

用户登录时，除了需要用户名和密码外，还需要填写随机生成的验证码。当验证码显示不清楚时，可以通过超链接进行更换而无须重新刷新页面。登录处理对登录进行了一些限制。例如，输入 3 次无效用户名后，将使用 Cookie 技术禁止该用户继续登录；当用户连续 3 次输入错误密码后，该账号将被冻结。用户登录流程如图 1.2 所示。





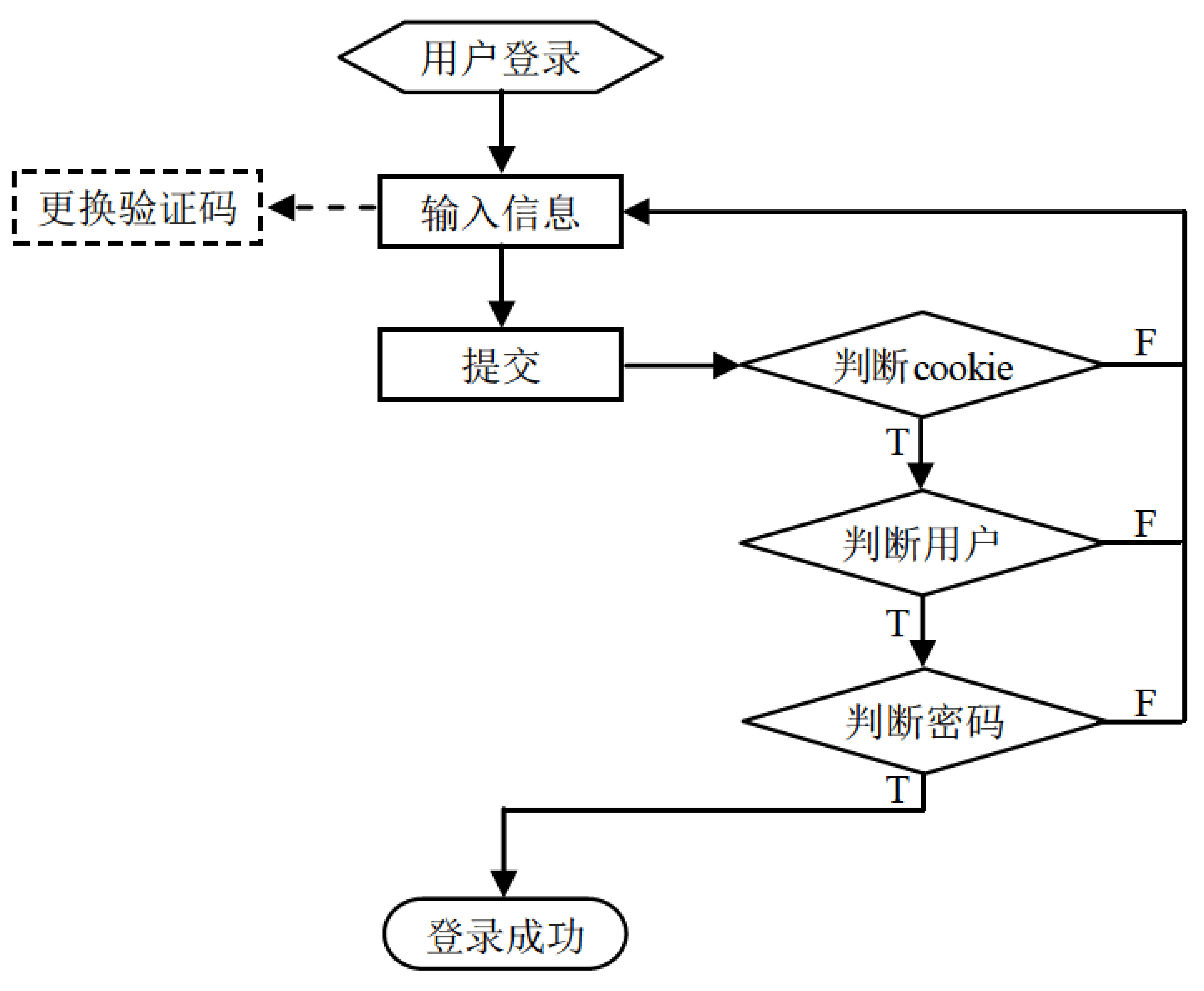



图 1.2　用户登录流程图

1.1.3　找回密码流程

当用户因为个别原因忘记密码时，可以通过密码保护问题及答案来找回。如果用户输入正确，那么系统会将密码通过电子邮件的形式发送到用户的注册邮箱中。如果没有填写密码保护问题及答案或者回答错误，则无法找回。找回密码的流程如图 1.3 所示。

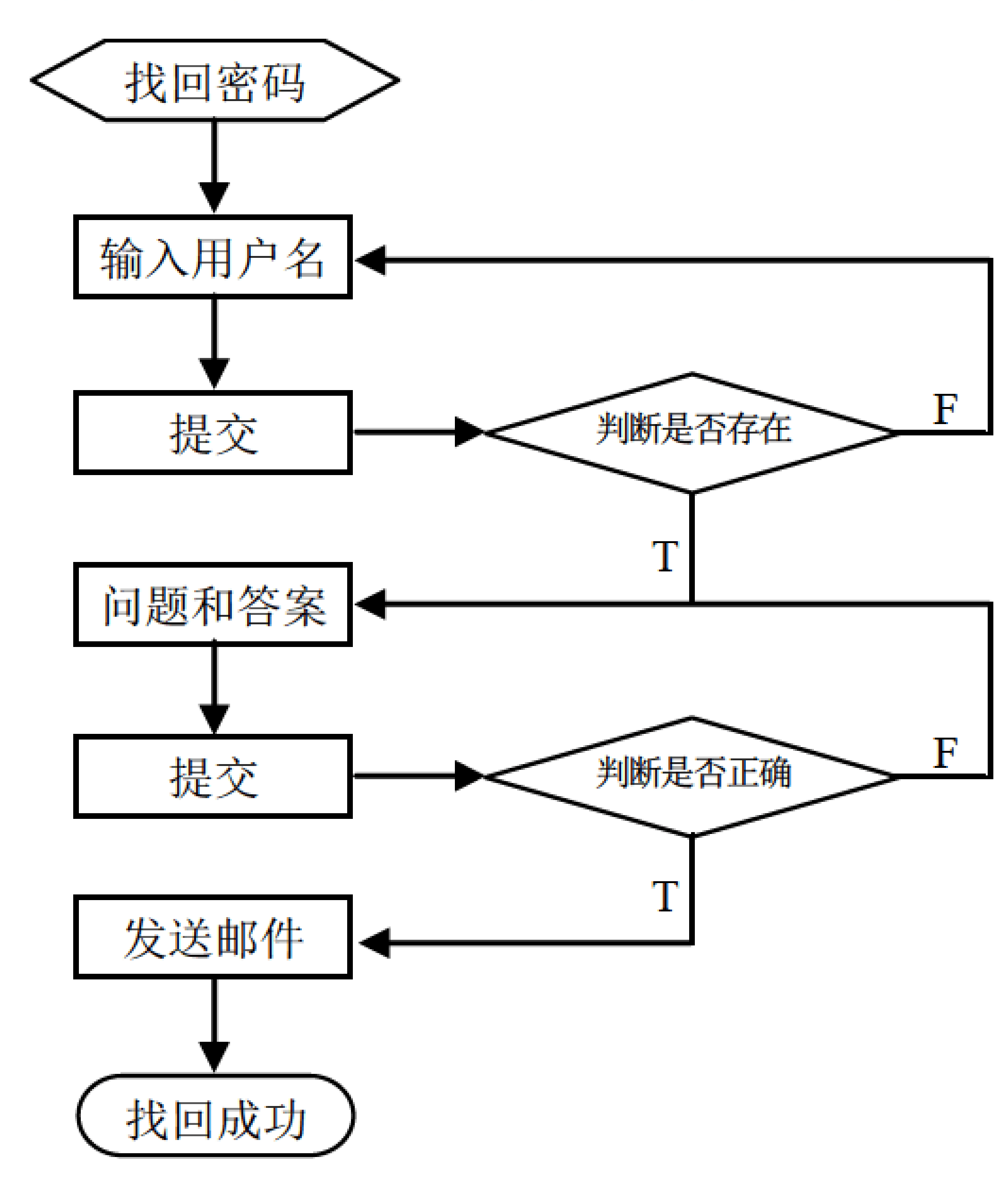


图 1.3　找回密码流程图



Note

1.1.4 程序预览

注册登录模块主要由两个页面组成，即注册页面和登录页面。注册页面与登录页面的运行效果如图 1.4 和图 1.5 所示。



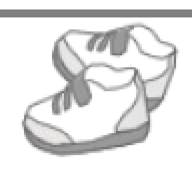
图 1.4 注册页面

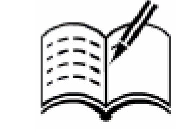


图 1.5 登录页面

1.2 数据库设计

本模块应用 MySQL 数据库存储用户注册信息，有关 MySQL 数据库及数据表的创建

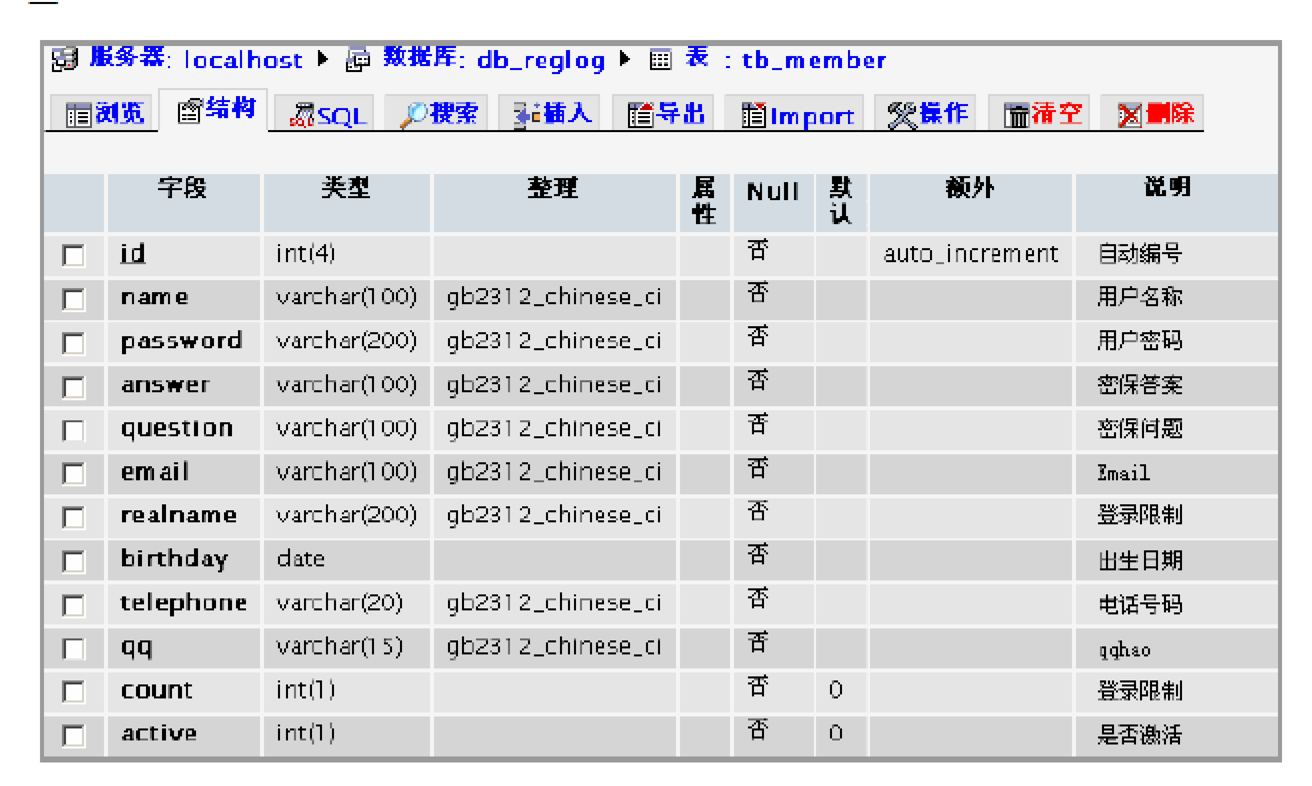




将在本节进行详细讲解。

1.2.1　数据库设计概述

实现注册登录模块使用的数据库是 db_reglog，其中包含一个数据表 tb_member，该表中有 12 个字段。tb_member 表的结构及说明如图 1.6 所示。



服务器: localhost ▸ 数据库: db_reglog ▸ 表 : tb_member

浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(100)	gb2312_chinese_ci		否			用户名称
password	varchar(200)	gb2312_chinese_ci		否			用户密码
answer	varchar(100)	gb2312_chinese_ci		否			密保答案
question	varchar(100)	gb2312_chinese_ci		否			密保问题
email	varchar(100)	gb2312_chinese_ci		否			Email
realname	varchar(200)	gb2312_chinese_ci		否			登录限制
birthday	date			否			出生日期
telephone	varchar(20)	gb2312_chinese_ci		否			电话号码
qq	varchar(15)	gb2312_chinese_ci		否			qqhao
count	int(1)			否	0		登录限制
active	int(1)			否	0		是否激活

图 1.6　tb_member 表结构及说明

1.2.2　封装数据库连接、操作类

为了便于维护，减少代码冗余，本模块使用了一个简单的自定义数据库类。下面给出类中的主要代码。

首先给出的是类中的成员变量，其中$host、$name、$pwd 和$dBase 变量是类初始化时所使用的，有默认值，其他变量都是相关的数据信息。代码如下：

```
<?php
class opmysql{
        private $host = 'localhost';                    //服务器地址
        private $name = 'root';                         //登录账号
        private $pwd = '111';                           //登录密码
        private $dBase = 'db_reglog';                   //数据库名称
        private $conn = '';                             //数据库连接资源
        private $result = '';                           //结果集
        private $msg = '';                              //返回结果
        private $fields;                                //返回字段
        private $fieldsNum = 0;                         //返回字段数
        private $rowsNum = 0;                           //返回结果数
        private $rowsRst = '';                          //返回单条记录的字段数组
        private $filesArray = array();                  /返回字段数组
        private $rowsArray = array();                   //返回结果数组
```

接下来给出的是构造函数。该构造函数包含 4 个参数，这 4 个参数都可以被省略而使用默认值。参数处理后，调用 init_conn()函数。代码如下：

```
        function __construct($host='',$name='',$pwd='',$dBase=''){
            if($host != '')
```




```
            $this->host = $host;
        if($name != '')
            $this->name = $name;
        if($pwd != '')
            $this->pwd = $pwd;
        if($dBase != '')
            $this->dBase = $dBase;
        $this->init_conn();
    }
```

init_conn()函数：根据成员变量中的值来创建数据库连接源。代码如下：

```
    //连接数据库
    function init_conn(){
        $this->conn=@mysql_connect($this->host,$this->name,$this->pwd);
        @mysql_select_db($this->dBase,$this->conn);
        mysql_query("set names gb2312");                    //设置编码
    }
```

下面开始就是类中的操作函数了。首先介绍查询函数。查询函数根据传过来的 SQL 语句进行查询，并将查询结果保存到成员变量$result 中。符号“@”的作用是屏蔽错误信息。代码如下：

```
    //查询结果
    function mysql_query_rst($sql){
        if($this->conn == ''){
            $this->init_conn();
        }
        $this->result = @mysql_query($sql,$this->conn);
    }
```

返回查询记录数函数：根据查询结果，返回记录数。代码如下：

```
    function getRowsNum($sql){
        $this->mysql_query_rst($sql);
        if(mysql_errno() == 0){
            return @mysql_num_rows($this->result);
        }else{
            return '';
        }
    }
```

取得记录数组函数：将查询结果输出成一个数组并返回。该函数处理的是单条记录。代码如下：

```
    function getRowsRst($sql){
        $this->mysql_query_rst($sql);
        if(mysql_error() == 0){
            $this->rowsRst = mysql_fetch_array($this->result,MYSQL_ASSOC);
            return $this->rowsRst;
        }else{
```



```
				return '';
			}
		}
```

取得记录数组函数：功能同上，只是这里返回一个有多条记录的二维数组。代码如下：

```
		//取得记录数组（多条记录）
		function getRowsArray($sql){
			$this->mysql_query_rst($sql);
			if(mysql_errno() == 0){
				while($row = mysql_fetch_array($this->result,MYSQL_ASSOC)) {
					$this->rowsArray[] = $row;
				}
				return $this->rowsArray;
			}else{
				return '';
			}
		}
```

返回更新、删除、添加的记录数函数：该函数用来更新、删除、添加记录并获取受影响的记录数，进而判断操作是否成功。代码如下：

```
		function uidRst($sql){
			if($this->conn == ''){
				$this->init_conn();
			}
			@mysql_query($sql);
			$this->rowsNum = @mysql_affected_rows();
			if(mysql_errno() == 0){
				return $this->rowsNum;
			}else{
				return '';
			}
		}
```

释放结果集函数：将不再使用的数据删除，释放内存。代码如下：

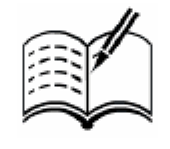
```
		function close_rst(){
			mysql_free_result($this->result);
			$this->msg = '';
			$this->fieldsNum = 0;
			$this->rowsNum = 0;
			$this->filesArray = '';
			$this->rowsArray = '';
		}
```

关闭数据库函数：关闭数据库。代码如下：

```
		//关闭数据库
		function close_conn(){
			$this->close_rst();
			mysql_close($this->conn);
```

Note

```
                $this->conn = '';
            }
    }
    $conne = new opmysql();
    ?>
```


Note

1.3 用 户 注 册

1.3.1 用户注册功能概述

用户注册时，必填信息包括用户名、密码和 E-mail。只有这些必填信息添加完整并正确时，“注册”按钮才被激活。这时用户可以选择注册或填写详细信息后再进行注册。在整个注册过程中，对于用户的每一次操作，系统都将给出友好的提示来帮助用户完成注册。注册功能的运行结果如图 1.7 所示。

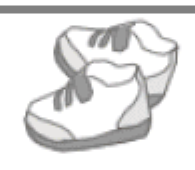


图 1.7　注册功能的运行结果

注册功能的实现涉及 5 个核心文件：register.php（注册表单页）、register_chk.php（注册信息处理页）、chkname.php（检测用户名）、register.js（注册信息验证脚本）和 activation.php（激活处理页）。

1.3.2 注册页面设计

在注册表单页中，主要包含填写信息的文本框、信息提示的 div 标签和 3 个相关按钮。注册表单页中使用的元素如表 1.1 所示。



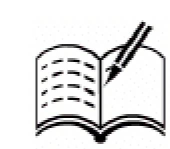


表 1.1　注册页面中的主要元素

元素名称	元素属性值	说　明
script	"js/xmlhttprequest.js"	载入xmlhttprequest对象脚本
script	"js/register.js"	载入注册信息验证脚本
text	id="regname" name="regname"	注册名称
div	id="namediv"	注册名称提示信息
password	id="regpwd1" name="regpwd1"	注册密码
div	id="pwddiv1"	密码提示信息
password	id="regpwd2" name="regpwd2"	确认密码
div	id="pwddiv2"	确认密码提示信息
text	id="email" name="email"	E-mail
div	id="emaildiv"	E-mail提示信息
div	id="morediv" style="display:none;"	隐藏/显示非必填信息
text	id=" question" name="question"	密码保护问题
text	id="answer" name="answer"	密码保护答案
text	id="realname" name="realname"	真实姓名
text	id="birthday" name="birthday"	出生日期
text	id="telephone" name="telephone"	联系电话
text	id="qq name=" qq	qq
button	id="regbtn" disabled="disabled"	“注册”按钮
button	id="morebtn"	显示/隐藏全部注册信息
button	id="logbtn"	返回登录页面
div	id="imgdiv" style=" visibility: hidden;"	等待进度条

当用户向注册页面中的表单元素输入信息时，将会触发注册信息验证脚本 register.js。下面逐行讲解 register.js 脚本页。

在 register.js 中，首先写了一个通用函数，来避免重复输入 document.getElementById('…')。这个函数的功能是返回被触发的“id=idValue”的元素对象。代码如下：

```
function $(idValue){
        return document.getElementById(idValue);
}
```

接下来是 window.onlaod 事件，表示当窗体被载入时触发。function(){…}表示当页面被载入时所要进行的操作。代码如下：

```
window.onload = function(){
        …
}
```

首先将焦点定位到用户名文本框，方便用户操作。接下来声明了 5 个变量，这 5 个变量代表了 5 项要检测的数据。当检测数据为“合格”时，代表它的变量将等于 yes。代码如下：





Note

```
$('regname').focus();                              //焦点定位到用户名文本框
//声明 5 个变量，表示要检测的 5 项数据
var cname1,cname2,cpwd1,cpwd2,cemail;
```

chkreg()函数是每一次触发键盘事件后都要调用的。该函数判断 5 个变量的值，只有当所有变量都等于 yes 时，“注册”按钮才会被激活。代码如下：

```
function chkreg(){
    if((cname1 == 'yes') && (cname2 == 'yes') && (cpwd1 == 'yes') && (cpwd2 == 'yes') && (cemail == 'yes')){
            $('regbtn').disabled = false;
        }else{
            $('regbtn').disabled = true;
        }
}
```

接下来是验证用户名。当用户输入注册名称时，该函数会把用户的每次输入都作一下正则判断，并根据结果设置不同的 cname1 值。代码如下：

```
//验证用户名
$('regname').onkeyup = function (){
    //获取注册名称
    name = $('regname').value;
    cname2 = '';
    if(name.match(/^[a-zA-Z_]*/) == ''){
        $('namediv').innerHTML = '<font color=red>必须以字母或下划线开头</font>';
        cname1 = '';
    }else if(name.length <= 3){
        $('namediv').innerHTML = '<font color=red>注册名称必须大于 3 位</font>';
        cname1 = '';
    }else{
        $('namediv').innerHTML = '<font color=green>注册名称符合标准</font>';
        cname1 = 'yes';
    }
    chkreg();
}
```

当用户名文本框失去焦点时，即用户输入完毕转到页面中其他元素的时候，将检测用户名是否重复。用户名判断采用了 Ajax 技术，调用 chkname.php 页，并根据 chkname.php 的返回值在 div 标签中显示判断结果。chkname.php 页的代码将稍后给出。

```
//当用户名文本框失去焦点时，系统调用处理页查看是否存在该用户
$('regname').onblur = function(){
    name = $('regname').value;                              //获取注册用户名
    //当用户名的格式输入合格后，才进行这一步操作
    if(cname1 == 'yes'){
        xmlhttp.open('get','chkname.php?name='+name,true);      //创建新请求
        xmlhttp.onreadystatechange = function(){
            if(xmlhttp.readyState == 4){
                if(xmlhttp.status == 200){
```

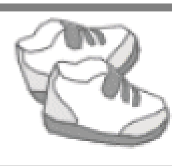





```
                    var msg = xmlhttp.responseText;          //获取响应页内容
                    if(msg == '1'){
                $('namediv').innerHTML="<font color=green>恭喜您，该用户名可以使用!</font>";
                        cname2 = 'yes';
                    }else if(msg == '2'){
                        $('namediv').innerHTML="<font color=red>用户名被占用!</font>";
                        cname2 = '';
                    }else{
                    $('namediv').innerHTML="<font color=red>"+msg+"</font>";
                        cname2 = '';
                    }
                }
            }
            chkreg();                              //检查是否要激活"注册"按钮
        }
        xmlhttp.send(null);
    }
}
```

接下来是验证密码。验证密码时，除了可以限制密码的长度外，还可以判断密码的强度。例如下述代码中，长度 6～12 之间的密码都是弱，如果长度大于 12，但密码单一，都是数字或文字，那么密码强度就为中，否则强度就为高。代码如下：

```
//验证密码
$('regpwd1').onkeyup = function(){
    pwd = $('regpwd1').value;
    pwd2 = $('regpwd2').value;
    if(pwd.length < 6){
        $('pwddiv1').innerHTML = '<font color=red>密码长度最少需要 6 位</font>';
        cpwd1 = '';
    }else if(pwd.length >= 6 && pwd.length < 12){
        $('pwddiv1').innerHTML = '<font color=green>密码符合要求。密码强度：弱</font>';
        cpwd1 = 'yes';
    }else if((pwd.match(/^[0-9]*$/)!=null) || (pwd.match(/^[a-zA-Z]*$/) != null )){
        $('pwddiv1').innerHTML = '<font color=green>密码符合要求。密码强度：中</font>';
        cpwd1 = 'yes';
    }else{
        $('pwddiv1').innerHTML = '<font color=green>密码符合要求。密码强度：高</font>';
        cpwd1 = 'yes';
    }
    if(pwd2 != '' && pwd != pwd2){
        $('pwddiv2').innerHTML = '<font color=red>两次密码不一致!</font>';
        cpwd2 = '';
    }else if(pwd2 != '' && pwd == pwd2){
        $('pwddiv2').innerHTML = '<font color=green>密码输入正确</font>';
        cpwd2 = 'yes';
    }
```



```
        chkreg();
    }
```

二次密码的验证比较简单，只要第一、第二次密码相等就可以。




```
    //验证二次密码
        $('regpwd2').onkeyup = function(){
        pwd1 = $('regpwd1').value;
        pwd2 = $('regpwd2').value;
        if(pwd1 != pwd2){
            $('pwddiv2').innerHTML = '<font color=red>两次密码不一致!</font>';
            cpwd2 = '';
        }else{
            $('pwddiv2').innerHTML = '<font color=green>密码输入正确</font>';
            cpwd2 = 'yes';
            chkreg();
        }
    }
```

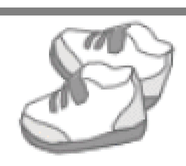
接下来的 E-mail 验证主要是对 E-mail 的格式进行检查。输入的字串中，必须包含“@”和“.”；同时，这两个符号的位置既不能在首和尾，也不能连在一起。代码如下：

```
    //验证 E-mail
    $('email').onkeyup = function(){
        emailreg = /^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$/;
        $('email').value.match(emailreg);
        if($('email').value.match(emailreg) == null){
            $('emaildiv').innerHTML = '<font color=red>错误的 email 格式</font>';
            cemail = '';
        }else{
            $('emaildiv').innerHTML = '<font color=green>输入正确</font>';
            cemail = 'yes';
        }
        chkreg();
    }
```

上面的部分是对必填项的检查。如果用户希望填写更详细的资料，可以单击“详细资料”按钮。代码如下：

```
    //显示/隐藏详细信息
    $('morebtn').onclick = function(){
        if($('morediv').style.display == ''){
            $('morediv').style.display = 'none';
        }else{
            $('morediv').style.display = '';
        }
    }
```

所有信息添加完毕并通过验证后，就可以注册了。用户单击“注册”按钮时，下述代码应用 Ajax 技术将获取的用户信息以 URL 的形式传给 register_chk.php 页，并根据返回结





Note

果作出相应的处理。代码如下：

```
//正式注册
$('regbtn').onclick = function(){
    $('imgdiv').style.visibility = 'visible';
    url = 'register_chk.php?name='+$('regname').value+'&pwd='+$('regpwd1').value+'&email='+$('email').value;
    url += '&question=' +$('question').value+'&answer='+$('answer').value;
    url += '&realname=' +$('realname').value+'&birthday='+$('birthday').value;
    url += '&telephone='+$('telephone').value+'&qq='+$('qq').value;
    alert(url);
    return false;
    xmlhttp.open('get',url,true);
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
            if(xmlhttp.status == 200){
                msg = xmlhttp.responseText;
                if(msg == '1'){
                    alert('注册成功，请到您的邮箱中获取激活码！');
                    location='main.php';
                }else if(msg == '-1'){
                    alert('您的服务器不支持 POP3，请仔细检查!!');
                }else{
                    alert(msg);
                }
            }
            $('imgdiv').style.visibility = 'hidden';
        }
    }
    xmlhttp.send(null);
}
```

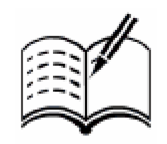
1.3.3　检测用户名是否被占用

检测用户名是否被占用需要用到 chkname.php 文件，其作用是检查用户名是否重复，并且返回判断结果。代码如下：

```
<?php
//载入数据库类
include_once "conn/conn.php";
//根据 name 查询 tb_member 表
$sql = "select * from tb_member where name='".$_GET['name']."'";
//获取查询结果的记录数
$num = $conne->getRowsNum($sql);
if($num == 1){                                  //如果有记录
    echo '2';
}else if($num == 0){                            //如果没有记录
    echo '1';
}else{
```



```
        echo $conne->msg_error();                  //否则出错
    }
    ?>
```



Note

1.3.4　注册信息处理

在注册信息处理页（register_chk.php）中，将 JavaScript 传递过来的值连成 insert 语句并执行。如果添加成功，则使用邮件函数发送一封激活邮件，并弹出提示对话框，如图 1.8 所示。

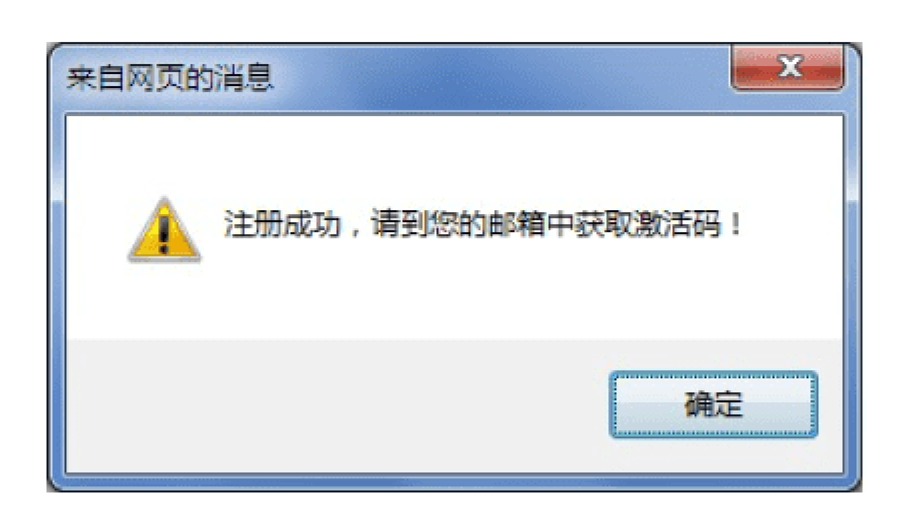


图 1.8　注册成功提示对话框

在处理代码中包含两种邮件的发送方法，一种是使用局域网，向指定 SMTP 服务器中的邮箱（这里的 cym3100@163.com 邮箱是在 SMTP 服务器中注册的测试邮箱）中发送邮件；另一种是使用互联网，通过指定的邮箱（mrsoft8888@sohu.com 是在搜狐网中注册的邮箱，用户名为 mrsoft8888，密码为 mrsoft8888）向用户提供的有效邮箱中发送邮件。处理页的代码如下：

```
<?php
    include_once 'conn/conn.php';
    require_once 'Zend/Mail.php';                              //调用发送邮件的文件
    require_once 'Zend/Mail/Transport/Smtp.php';               //调用 SMTP 验证文件
    $reback = '0';
    $url = 'http://'.$_SERVER['SERVER_NAME'].dirname($_SERVER['SCRIPT_NAME']).'/activation.php';
    $url .= '?name='.trim($_GET['name']).'&pwd='.md5(trim($_GET['pwd']));
    //发送激活邮件
    $subject="激活码的获取";
    $mailbody='注册成功。您的激活码是：
'.'<a href="'.$url.'" target="_blank">'.$url.'</a><br>'.'请点击该地址，激活您的用户！';
    //定义邮件内容
    $envelope="mrsoft8888@sohu.com";                           //定义登录使用的邮箱
/*  SMTP 测试版发送邮件方法，使用 SMTP 作为服务器
    $tr = new Zend_Mail_Transport_Smtp('192.168.1.247');
    $mail = new Zend_Mail();
    $mail->addTo('cym3100@163.com','获取用户注册激活码');
    $mail->setFrom('cym3100@163.com','明日科技典型模块程序测试邮箱，恭喜您用户注册成功！');
    $mail->setSubject('获取注册用户的激活码');
    $mail->setBodyHtml($mailbody);
```






```
        $mail->send($tr);
    */
    /*  网络版发送邮件方法  */
        $config = array('auth' => 'login',
                'username' => 'mrsoft8888',
                'password' => ' mrsoft8888');                 //定义 SMTP 的验证参数
        $transport = new Zend_Mail_Transport_Smtp('smtp.sohu.com', $config);
                                                              //实例化验证的对象
        $mail = new Zend_Mail('GBK');                         //实例化发送邮件对象
        $mail->setBodyHtml($mailbody);                        //发送邮件主体
    $mail->setFrom($envelope, '明日科技典型模块程序测试邮箱,恭喜您用户注册成功！');
                                                              //定义邮件发送使用的邮箱
        $mail->addTo($_GET[email], '获取用户注册激活码');     //定义邮件的接收邮箱
        $mail->setSubject('获取注册用户的激活码');            //定义邮件主题
        $mail->send($transport);                              //执行发送操作
    /*  网络版发送邮件方法  */
        $sql = "insert into tb_member(name,password,question,answer,email,realname,birthday,
telephone,qq)
    values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."','".$_
GET['answer']."',
    '".$_GET['email']."','".$_GET['realname']."','".$_GET['birthday']."','".$_GET['telephone']."','
".$_GET['qq']."')";
        $num = $conne->uidRst($sql);
        if($num == 1){
            $reback = '1';
        }
        echo $reback;
    ?>
```

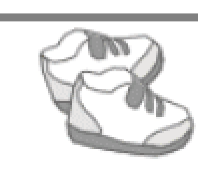
1.3.5　注册用户激活

在注册成功后，用户需要登录自己填写的邮箱，浏览注册时发送的邮件，并且单击超链接，链接到 activation.php 文件，根据超链接传递的用户名和密码完成对指定用户表中 active 字段值的更新，即完成注册用户的激活操作。

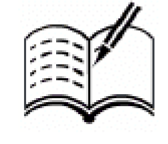
登录 mrsoft8888@sohu.com 邮箱，将查看到如图 1.9 所示的邮件内容。



图 1.9　激活邮件的运行结果



当用户进入激活页（activation.php）后，激活页将根据 name 字段和 pwd 字段来查找匹配的注册用户，如果找到将进行更新操作。激活成功后，将用户名保存到 Session 中，并跳转到相应的页面。代码如下：



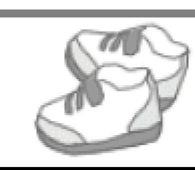
Note

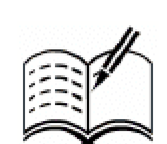
```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
include_once("conn/conn.php");
if (!empty($_GET['name']) && !is_null($_GET['name'])){                    //激活注册用户
        $num=$conne->getRowsNum("select * from tb_member
where name='".$_GET['name']."' and password = '".$_ GET['pwd']."'");
        if ($num>0){
            $upnum=$conne->uidRst("update tb_member set active = 1
where name='".$_GET['name']."' and password = '".$_GET['pwd']."'");
            if($upnum > 0){
                $_SESSION['name'] = $_GET['name'];
                echo "<script>alert('用户激活成功！');window.location.href= 'main.php'; </script>";
            }else{
                echo "<script>alert('您已经激活！');window.location.href='main.php'; </script>";
            }
        }else{
            echo "<script>alert('用户激活失败！');window.location.href='register.php'; </script>";
        }
}
?>
```

1.3.6 免激活用户注册

上面介绍的用户注册功能中，需要将激活信息以邮件的形式发送到用户指定的邮箱中进行激活后，用户才可以使用。这里向读者介绍一种不需要激活的用户注册方式，同样以上面注册提交的数据为基础，只需修改用户注册提交数据的处理方式，就可以实现免激活的用户注册功能。该操作在 register_check.php 文件中完成。其关键代码如下：

```
<?php
    include_once 'conn/conn.php';                    //包含数据库连接操作文件
    $reback = '0';                                   //定义初始变量
    $sql = "insert into tb_member(name,password,question,answer,email,realname,birthday,
telephone,qq,active)
values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."','".$_
GET['answer']."',
'".$_GET['email']."','".$_GET['realname']."','".$_GET['birthday']."','".$_GET['telephone']."','
".$_GET['qq']."','1')";
        $num = $conne->uidRst($sql);                 //执行添加语句
        if($num == 1){
            $reback = '1';
        }
    echo $reback;                                    //输出变量值
?>
```





说明：

在本模块中要使用免激活模式来注册用户，首先应该在 js\register.js 文件中修改注册数据处理 URL 路径，将注册数据处理文件（register_chk.php）修改为 register_check.php。代码如下：

```
$('regbtn').onclick = function(){
        $('imgdiv').style.visibility = 'visible';
        url = 'register_chk.php?name='+$('regname').value+'&pwd='+$('regpwd1').value+'&email z='+$('email').value;
        url += '&question=' +$('question').value+'&answer='+$('answer').value;
        url += '&realname=' +$('realname').value+'&birthday='+$('birthday').value;
        url += '&telephone='+$('telephone').value+'&qq='+$('qq').value;
        xmlhttp.open('get',url,true);
        xmlhttp.onreadystatechange = function(){
            if(xmlhttp.readyState == 4){
                if(xmlhttp.status == 200){
                    msg = xmlhttp.responseText;
                    if(msg == '1'){
                        alert('注册成功，请到您的邮箱中获取激活码！');
                        location='index.php';
                    }else if(msg == '-1'){
                        alert('您的服务器不支持 Zend_mail，或者邮箱填写错误。请仔细检查!! ');
                    }else{
                        alert(msg);
                    }
                    $('imgdiv').style.visibility = 'hidden';
                }
            }
        }
        xmlhttp.send(null);
}
```

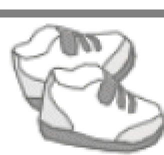
如此一来，在执行用户注册数据的处理时，就会调用上面重新编写的 register_check.php 文件，用户注册成功后可以直接登录，而不必再进行激活操作。

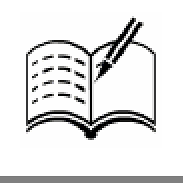
1.4　用户登录

1.4.1　用户登录功能概述

用户登录功能即对用户输入的账号、密码以及验证码进行验证。验证内容包括：信息是否为空、验证码是否错误、用户计算机是否已经被锁定、用户是否存在、用户是否激活、用户名和密码是否匹配、用户登录次数是否达到上限等。登录功能的运行结果如图 1.10 所示。

登录功能包含 3 个核心文件，分别是 login.php（用户登录页）、login.js（登录信息验证脚本）和 login_chk.php（登录处理页）。





Note



图 1.10　单击“登录”按钮后的运行结果

1.4.2　登录页面设计

1. 用户登录页（以下简称登录页）

相对于注册页面来说，登录页（login.php）中的表单元素要少很多，也没有那么复杂。登录页中使用的元素如表 1.2 所示。

表 1.2　登录页中的元素

元 素 名 称	元素属性值	说　　明
script	"js/xmlhttprequest.js"	载入xmlhttprequest对象脚本
script	"js/login.js"	载入登录信息验证脚本
text	id="lgname" name="lgname"	注册名称
password	id="lgpwd" name="lgpwd"	注册密码
text	id="lgchk" name="lgchk"	验证码
img	id='chkid' src="" width="55" height="18"	图形验证码
a	id="changea"	更换验证码超链接
hidden	id = "chknm" name = "chknm" value=""	隐藏域，保存验证码
button	id="lgbtn"	“登录”按钮
button	id="reg"	跳转到注册页
button	id="found"	找回密码

2. 登录信息验证脚本（login.js）

在登录页 login.php 中，载入了 login.js 脚本文件，首先通过该脚本文件完成对登录用户的所有验证操作。代码如下：

```
function $(id){
        return document.getElementById(id);
}
window.onload = function(){
        …
}
```





Note

接下来的函数功能是焦点自动下移。当焦点移动到最后一个文本框即验证码文本框后，再次按下 Enter 键，将调用 chklg()函数对用户输入的信息进行验证。实现原理将在后面的“技术提炼”中进行详细讲解。代码如下：

```
$('lgname').focus();                          //光标定位到用户名文本框
$('lgname').onkeydown = function(){           //处理用户名文本库的键盘按下事件
    if(event.keyCode == 13){                  //当按下 Enter 键时，光标自动下移
        $('lgpwd').select();                  //密码文本框获取焦点
    }
}
$('lgpwd').onkeydown = function(){
    if(event.keyCode == 13){                  //处理密码文本框的鼠标按下事件
        $('lgchk').select();                  //验证码文本框获得焦点
    }
}
$('lgchk').onkeydown = function(){            //处理验证码文本框的鼠标按下事件
    if(event.keyCode == 13){
        chklg();                              //调用 chklg()函数
    }
}
$('lgbtn').onclick = chklg;
```

然后进行最基本的检测，包括用户名是否合法、信息不能为空、验证码是否输入错误等。代码如下：

```
function chklg(){
        /*判断用户名是否合法，用户名第一位必须是英文字母，不允许有特殊字符*/
        if($('lgname').value.match(/^[a-zA-Z_]\w*$/) == null){
            alert('请输入合法名称');
            $('lgname').select();
            return false;
        }
        /*  用户名是否为空  */
        if($('lgname').value == ''){
            alert('请输入用户名!');
            $('lgname').focus();
            return false;
        }
        /*  密码是否为空  */
        if($('lgpwd').value == ''){
            alert('请输入密码!');
            $('lgpwd').focus();
            return false;
        }
        /*  验证码是否为空  */
        if($('lgchk').value == ''){
            alert('请输入验证码');
            $('lgchk').select();
            return false;
```

```
        }
        /*   验证码输入是否正确   */
        if($('lgchk').value != $('chknm').value){
                alert('验证码输入错误');
                $('lgchk').select();
                return false;
        }
```

接下来，判断客户端的 Cookie 值是否大于 3，如果大于 3，说明用户是非法操作的，将停止继续执行。代码如下：

```
        count = document.cookie.split(';')[0];                  //获取 Cookie 信息
        if(count.split('=')[1] >= 3){
                alert('因为您的非法操作，您将无法再执行登录操作');
                return false;
        }
```

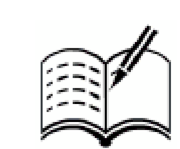
最后，将用户输入的信息传递给 login_chk .php 页进行处理。代码如下：

```
        url  =  'login_chk.php?act='+(Math.random())+'&name='+$('lgname').value+'&pwd='+$('lgpwd').value;
        xmlhttp.open('get',url,true);
        xmlhttp.onreadystatechange = function(){
                if(xmlhttp.readystate == 4){
                        if(xmlhttp.status == 200){
                                //获取 login_chk.php 页的返回值
                                msg = xmlhttp.responseText;
                                if(msg == '0'){
                                    alert('您还没有激活，请先登录邮箱进行激活操作。');
                                }else if(msg == '1'){
                                    alert('用户名或密码输入错误，您还有 2 次机会');
                                    $('lgpwd').select();
                                }else if(msg == '2'){
                                    alert('用户名或密码输入错误，您还有 1 次机会');
                                    $('lgpwd').select();
                                } else if(msg == '4'){
                                    alert('用户名输入错误');
                                    $('lgname').select();
                                } else if(msg == '-1'){
                                    alert('登录成功');
                                    location = 'main.php';
                                }else{
                                    alert(msg);
                                }
                        }
                }
        }
        xmlhttp.send(null);
    }
```

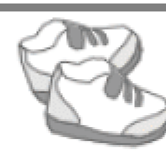


1.4.3　登录处理页

在登录处理页（login_chk.php）中，对表单中提交的用户名、密码进行验证，包括用户名是否存在、用户名是否被激活、用户登录次数、密码是否正确等，最终返回验证结果。代码如下：




```
<?php
    session_start();                                          //开启 Session 支持
    header('Content-Type:text/html;charset=gb2312'); //设置头信息
    include_once 'conn/conn.php';                             //载入数据库类
    $name = $_GET['name'];
    $pwd = $_GET['pwd'];                                      //获取名称和密码
    if(!empty($name) and !empty($pwd)){
        //得到当前数据表中的 count 字段和 active 字段
        $sql = "select name,count,active from tb_member where name = '".$name."'";
        $active = $conne->getFields($sql,2);
        $count = $conne->getFields($sql,1);
        $conne->close_rst();
        //判断用户是否存在
        if($active == ''){
            if(is_null($_COOKIE['count']) or $_COOKIE['count'] == 0){
                setcookie('count',1);
            }else{
                setcookie('count',$_COOKIE['count']+1);
            }
            $reback = 4;
        }else if($active == 0){                               //判断用户是否被激活
            $reback = '0';
        }else if($count >= 3){
            $reback = '3';                                    //判断用户的登录次数
        }else{
            $sql .= " and password = '".md5($pwd)."'";
            $num = $conne->getRowsNum($sql);
            if($num == 0 or $num == ''){                      //如果用户密码错误，登录次数加 1
          $num = $conne->uidRst("update tb_member set count = ".($count+1)." where name = '".$name."'");
                $reback = ($count+1);
            }else{                                            //登录成功，清空 count 字段值
                if($count != 0){
                    $num = $conne->uidRst("update tb_member set count = 0 where name = '".$name."'");
                }
                //设置 cookie
                if(isset($_COOKIE['count']) and $_COOKIE['count'] != 0){
                    setcookie('count',0);
                }
                $_SESSION['name'] = $name;
                $reback = '-1';
            }
        }
    }
```



```
        echo $reback;
?>
```



1.4.4　生成及刷新验证码

验证码生成应用的是 GD2 函数库中的函数，操作存储于 valcode.php 文件中，其完整代码参见 1.6.3 节的内容，这里不再赘述。

当页面第一次被载入时，将执行 login.js 中的 showval()函数来生成验证码，并且调用 valcode.php 文件以图像形式输出验证码。因为验证码中带有干扰串，如果用户看不清，可以单击首页中的“看不清”超链接来再次执行 shwoval()函数，即刷新验证码。login.js 脚本中 showval()函数的代码如下：

```
showval();                                        //调用 showval()函数生成图文验证码
$('changea').onclick = showval;                   //更换验证码
//验证码生成函数
        function showval(){
            num = '';
            for(i=0;i<4;i++){
                 …                                //生成随机数
            }
            $('chkid').src='valcode.php?num='+num; //显示图文验证码
            $('chknm').value = num;               //将验证码保存到 chnm 隐藏域
        }
```

1.5　找回密码

1.5.1　找回密码功能概述

找回密码功能用到的技术也是邮箱技术。用户输入正确的账号、密保问题、密保答案后，系统将随机生成的一组数字发送到用户的邮箱中，用户可以通过邮箱来得到新密码。找回密码功能的运行结果如图 1.11 所示。

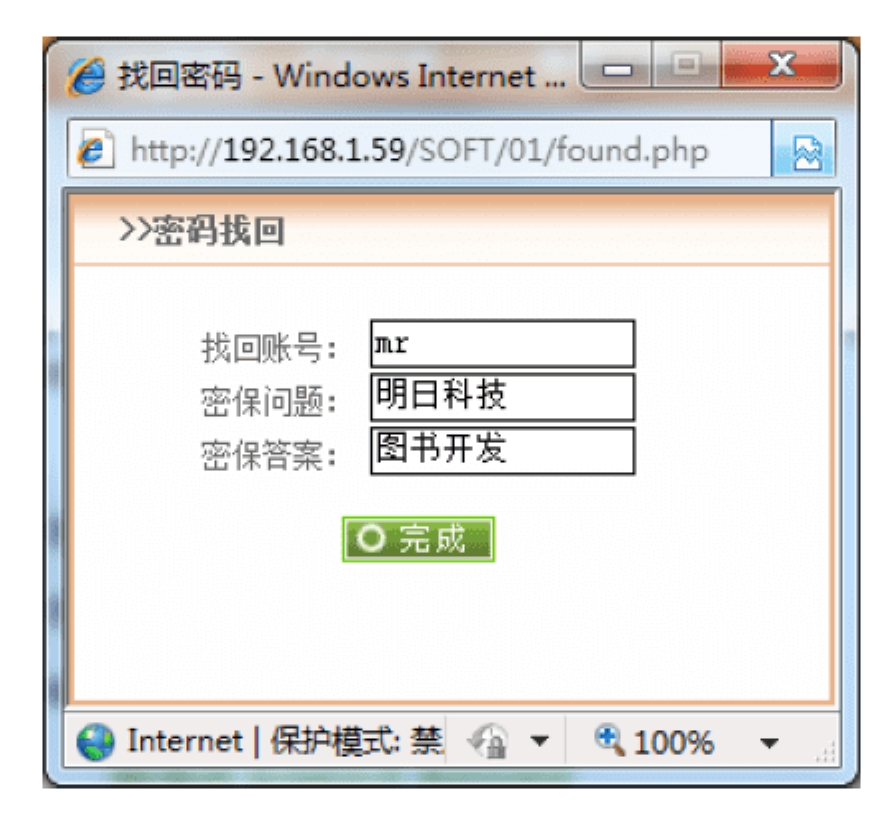


图 1.11　找回密码功能的运行结果





Note

1.5.2　找回密码功能的实现过程

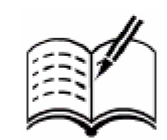
找回密码操作从单击登录页面中的“找回密码”按钮开始，在弹出的 found.php 文件中创建一个表单，提交找回账号、密保问题和密保答案，并且载入 xmlhttprequest.js 和 found.js 脚本文件。found.php 的关键代码如下：

```
<link rel="stylesheet" href="css/style.css" />
<script language="javascript" src="js/found.js"></script>
<script language="javascript" src="js/xmlhttprequest.js"></script>
<div id="fdbgdiv" >
	<div id="top"> >>密码找回</div>
	<div id="foundnamediv">找回账号：
<input id="foundname" type="text" style=" width: 100px; height:15px; border:1px #000000 solid;" /></div>
	<div id="foundnamediv">密保问题：
<input id="fdquestion" type="text" style=" width: 100px; height:15px; border:1px #000000 solid;" /></div>
  <div id="foundnamediv">密保答案：
<input id="fdanswer" type="text" style=" width: 100px; height:15px; border:1px #000000 solid;" /></div>
  <div id="foundnamediv" align="center"><button id="step1"></button></div>
</div>
```

在 found.js 文件中，调用 found_chk.php 文件，完成密码的找回操作。其关键代码如下：

```
function $(id){
	return document.getElementById(id);
}
window.onload = function(){
	$('foundname').focus();
	$('step1').onclick = function(){
		if($('foundname').value != '' && $('fdquestion').value != '' && $('fdanswer').value != ''){
			xmlhttp.open('get','found_chk.php?foundname='+$('foundname').value+'&question='+$('fdquestion').value+'&answer='+$('fdanswer').value,true);
			xmlhttp.onreadystatechange = function(){
				if(xmlhttp.readyState == 4 && xmlhttp.status == 200){
					msg = xmlhttp.responseText;
					if(msg == '1'){
						alert('找回密码成功，请登录注册邮箱!');
						window.close();
					}else{
						alert('填写信息错误！ ');
					}
				}
			}
			xmlhttp.send(null);
		}else{
			alert('请填写完整信息');
```





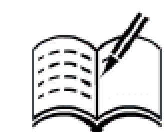
Note

```
                $('foundname').focus();
                return false;
            }
        }
    }
```

在 found_chk.php 文件中，对用户提交的密保问题和密保答案进行验证，如果正确则生成新的密码，并且更新数据库中存储的密码，同时将新密码发送到用户注册时填写的邮箱中。found_chk.php 的关键代码如下：

```
<?php
    include_once 'conn/conn.php';
    require_once 'Zend/Mail.php';                          //调用发送邮件的文件
    require_once 'Zend/Mail/Transport/Smtp.php';           //调用 SMTP 验证文件
    $reback = '0';
    $name = $_GET['foundname'];                            //获取用户名
    $question = $_GET['question'];                         //获取密码保护问题
    $answer = $_GET['answer'];                             //获取密码保护的答案
    $sql = "select email from tb_member
where name = '".$name."' and question = '".$question."' and answer = '".$answer."'";    // 定 义 查询语句
    $email = $conne->getFields($sql,0);                    //判断用户提交的用户密保是否正确
    if($email != ''){                                      //如果正确则执行如下操作
        $rnd = rand(1000,time());                          //生成随机数
        $sql = "update tb_member set password = '".md5($rnd)."'
where name = '".$name."' and question = '".$question."' and answer = '".$answer."'";    // 更 新 密码
        $tmpnum = $conne->uidRst($sql);                    //执行更新操作
        if($tmpnum >= 1){
            //发送密码邮件
            $subject="找回密码";
            $mailbody='密码找回成功。您账号的新密码是'.$rnd;
            $envelope["from"]="cym3100@163.com";
            //$envelope="mrsoft8888@sohu.com";             //网络版定义登录使用的邮箱
    /*  SMTP 测试版发送邮件方式，使用 SMTP 作为服务器*/
            $tr = new Zend_Mail_Transport_Smtp('192.168.1.247');
            $mail = new Zend_Mail();
            $mail->addTo($email,'获取用户新密码');
            $mail->setFrom('cym3100@163.com','明日科技典型模块程序测试邮箱，修改用户注册密码！');
            $mail->setSubject($subject);
            $mail->setBodyHtml($mailbody);
            //$mail->send($tr);
    /*  网络版发送邮件方法  */
    /*$config = array('auth' => 'login',
            'username' => 'mrsoft8888',
            'password' => ' mrsoft8888');              //定义 SMTP 的验证参数
    $transport = new Zend_Mail_Transport_Smtp('smtp.sohu.com', $config);        // 实 例
```






```
化验证的对象
        $mail = new Zend_Mail('GBK');              //实例化发送邮件对象
        $mail->setBodyHtml($mailbody);             //发送邮件主体
        $mail->setFrom($envelope, '明日科技典型模块程序测试邮箱，修改用户注册密码！');
        $mail->addTo($email, '获取用户新密码');    //定义邮件的接收邮箱
        $mail->setSubject($subject);               //定义邮件主题
        $mail->send($transport);                   //执行发送操作*/
    /*  网络版发送邮件方法  */
        if(false ==$mail->send($tr) ){
            $reback = '-1';
        }else{
            $reback = '1';
        }
    }else{
        $reback = '2';
    }
    }else{
        $reback = $sql;
    }
        echo $reback;
    ?>
```

1.6 技 术 提 炼

1.6.1 防 SQL 注入技术

SQL 注入，是指一些精通 SQL 语句的用户，通过在表单或浏览器地址栏中输入 SQL 语句来绕过系统检验的一种技术。一般防范 SQL 注入的手段是过滤敏感字符，例如引号等。

相对于 ASP 来说，PHP 要安全得多，但这不代表 PHP 网站就不会出现 SQL 注入的情况。幸运的是，在 PHP 上防 SQL 注入，要比 ASP 简单、方便，不需要编写一大段的转换语句，只要几个函数就可以。

（1）intval()函数

intval()函数的作用是返回变量的整数值，其语法如下：

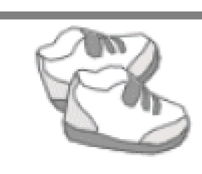
```
int intval ( mixed var [, int base] )
```

（2）addslashes()函数

addslashes()函数用于在操作数据库时，对其中的特殊字符进行自动转义，即在特殊字符前加上反斜杠（\），包括单引号（'）、双引号（"）、NULL，但是不包括“%”和“_”。其语法格式如下：

```
string addslashes ( string str )
```

此外，还可以使用 mysql_real_escape_string()来进行转义，效果和 addslashes()函数是一样的。



1.6.2　Ajax 无刷新验证技术



Note

Ajax（Asynchronous JavaScript and XML，异步 JavaScript 和 XML）是时下最流行的技术。它并不是新的技术，而是原有技术的集合，这从它的名称上就能够看出来。

Ajax 的核心技术是 XMLHttpRequest。通过 XMLHttpRequest 中的 open()方法和 send()方法，可以在不刷新当前页面的情况下向处理页发送数据；通过 XMLHttpRequest 中的 responseText 属性和 responseXML 属性，可以得到处理页的输出结果。

Ajax 之所以能够流行，是因为它能为用户提供良好的交互性。这个特性在本模块中被充分地体现出来，注册用户无须经过“漫长”的等待就可以知道输入的账号是否可用，登录用户也不必担心因为输错登录信息而重新刷新整个页面。

使用 Ajax，一般分为下面几步。

（1）首先要创建 XMLHttpRequest 对象。不同的浏览器，创建 XMLHttpRequest 对象及使用的方法有一些差别。这里只针对 IE 浏览器进行创建，代码如下：

```
var xmlhttp = false;                                    //初始化变量
//如果 ActiveXObject 存在，说明是 IE 1.0 以上的版本，否则使用 XMLHttpRequest 创建
if(window.ActiveXObject){
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}else if(window.XMLHttpReuqest){
        xmlhttp = new XMLHttpRequest();
}
```

（2）对象创建成功后，就可以使用对象中的 open()方法创建新请求了。其语法格式如下：

```
xmlhttp.open(rmethod,rurl,isAsync);
```

参数说明如下。

☑　rmethod：指明请求的方法，如 get()或 post()。

☑　rurl：指明请求的页面。可以是绝对地址，也可以是相对地址。

☑　isAsync：指明请求是否为异步。默认为 True，即异步。

（3）如果 isAsync 等于 True，那么当请求的状态改变时，将调用 onreadystatechange 属性，该属性指定了一个回调函数。其语法格式如下：

```
xmlhttp.onreadystatechange = reabackfunc;
```

或者

```
xmlhttp.onreadystatechange = function(){…}
```

（4）在回调函数中，首先需要判断 HTTP 的请求状态和 HTTP 状态码。这是通过 readyState 属性和 status 属性来判断的。readyState 属性有 5 种状态值，常用的是 4，表示数据接收完毕。status 属性的值比较多，常用的是 200，表示请求成功。一般这两个属性一起来判断。其语法格式如下：

```
xmlhttp.onreadystatechange = function(){
        if(readysate == 4 and statues == 200){
```






```
            ...
        }
    }
```

（5）当响应页处理结束后，即满足了“readystate==4 and status == 200”这个条件，就可以使用 xmlhttprequest 对象的属性获取响应页的值了。常用的有 responseText、responesXML、responseStream 等，这里以 responseText 为例进行介绍。

responesText 属性是将响应页的输出信息作为字符串返回。其语法格式如下：

```
str = xmlhttp.responseText;
```

（6）最后使用 send()方法来接收回应。send()方法可以传递数据，但这取决于 open()方法中的 rmethod 参数，当参数值为 get()时，数据是附在 URL 中进行传递的；当参数值为 post()时，数据只能使用 send()方法进行传递。send()方法的语法格式如下：

```
xmlhttp.send([rdate]);
```

1.6.3　验证码技术

验证码技术，是为了防止用户名和密码被暴力破解，在登录页面上生成一组随机数，每次刷新页面时，随机数都会改变。一般是 4 位，也有更多位的。

1. 获取验证码

PHP 中的验证码可以通过 rand()函数生成随机数的方式得到。rand()函数可以获取指定范围内的随机数，其语法格式如下：

```
int rand([int min, int max])
```

如果省略两个参数，那么将返回 0～RAND_MAX 之间的随机整数；否则，返回 min～max 之间的整数。例如，本例中要获取 4 位十六进制的整数，代码如下：

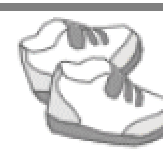
```
<?php
    for($i=0;$i<4;$i++){
        $num .= dechex(rand(0,15));                    //生成随机数
    }
    …
?>
```

函数 dechex()可以将参数转换为十六进制表示。

使用 JavaScript 也可以生成十六进制随机数，但是稍有些复杂。JavaScript 中不能直接将十进制数转换为十六进制，需要手动进行转换。首先使用 Math.random()函数生成 0～15 之间的随机数，然后使用 Math.ceil()函数将随机数取整，接下来就要逐次判断该值，如果该值大于 9，那么将 10～15 的数一一对应转换为 a、b…，一直到 f。转换完成后，将值累加，最后传给 valcode.php 页。

使用 JavaScript 生成十六进制随机数的完整代码如下：

```
//生成随机数
```




```
	function showval(){
		num = '';
		for(i=0;i<4;i++){						//循环输出 4 位验证码
			tmp =	Math.ceil((Math.random() * 15));	//取得 1 位十六进制的整数
				if(tmp > 9){					//依次判断随机数
					switch(tmp){
						case(10):			//如果随机数等于 10，转换为 a
							num += 'a';
							break;
						case(11):
							num += 'b';		//如果随机数等于 11，转换为 b
							break;
						case(12):
							num += 'c';		//如果随机数等于 12，转换为 c
							break;
						case(13):			//如果随机数等于 13，转换为 d
							num += 'd';
							break;
						case(14):			//如果随机数等于 14，转换为 e
							num += 'e';
							break;
						case(15):			//如果随机数等于 15，转换为 f
							num += 'f';
							break;
					}
				}else{
					num += tmp;
				}
		}
		$('chkid').src='valcode.php?num='+num;		//将生成的随机数传给图像生成页
		$('chknm').value = num;					//将随机数的值保存到页面的隐藏域中
	}
```

2. 显示随机数图片

显示随机数的方式很多，将随机数写入一个图片中再显示是目前比较常用的一种。在PHP中，可以使用GD函数库来实现。其中用到的函数主要有imagecreate()函数、imagecolorallocate()函数、imagestring()函数、imagesetpixel()函数、imagepng()函数和imagedestroy()函数。

（1）imagecreate()函数

imagecreate()函数用来创建一个基于调色板的空白图像源，这是生成图片的第一步。其语法格式如下：

```
resource imagecreate ( int width, int height )
```

参数 width 和 height 分别指定了图像的宽和高。

（2）imagecolorallocate()函数

imagecolorallocate()函数可以为创建后的图像分配颜色。其语法格式如下：

```
int imagecolorallocate ( resource image, int red, int green, int blue )
```





Note

参数说明如下。

- ☑ image：是一个图像源。
- ☑ red、green 和 blue：表示红、黄、蓝三元素的成分，每种颜色的取值范围在 1～255 之间。

（3）imagestring()函数

图像创建完成后，就可以使用 imagestring()函数来添加图像文字了。其语法格式如下：

```
bool imagestring ( resource image, int font, int x, int y, string s, int col )
```

参数说明如下。

- ☑ image：是一个图像源。
- ☑ font：用于设置字体，如果使用系统默认字体，可以使用 1～5 的数字。
- ☑ x 和 y：分别表示文字相对于整幅图像的 x 轴和 y 轴坐标，即所输入的字符串的左上角坐标。
- ☑ s：要显示的字符串。
- ☑ col：字体颜色，也是使用 imagecolorallocate()函数来分配。

（4）imagesetpixel()函数

使用 imagecolorallocate()函数创建的是一个单一背景色的图像，如果希望向图像中添加干扰码，可以使用 imagesetpixel()函数，该函数的作用是画一个像素点。其语法格式如下：

```
bool imagesetpixel ( resource image, int x, int y, int color )
```

参数说明和 imagestring()函数相似，这里不再赘述。

（5）imagepng()函数

该函数将创建完成的图片以 PNG 的格式输出。其语法格式如下：

```
bool imagepng ( resource image [, string filename] )
```

参数说明如下：

- ☑ image：要保存的图像源。
- ☑ filename：要保存的图像名。如果省略，则直接输出到浏览器。

（6）imagedestroy()函数

图像保存完毕后，使用 imagedestroy()函数来释放内存。其语法格式如下：

```
bool imagedestroy ( resource image )
```

验证码的生成文件是 valcode.php，其完整代码如下。

```
<?php
	header("content-type:image/png");		//设置页面编码
	$num = $_GET['num'];				//获取超链接传递的随机数
	$imagewidth=60;					//定义画布的宽
	$imageheight=18;				//定义画布的高
	$numimage = imagecreate($imagewidth,$imageheight);		//创建画布
	imagecolorallocate($numimage,240,240,240);			//设置画布颜色
	for($i=0;$i<strlen($num);$i++){					//循环读取随机数
```





```
        $x = mt_rand(1,8)+$imagewidth*$i/4;
        $y = mt_rand(1,$imageheight/4);
    $color=imagecolorallocate($numimage,mt_rand(0,150),mt_rand(0,150),mt_rand(0,150));
//定义图像的颜色
        imagestring($numimage,5,$x,$y,$num[$i],$color);  //将随机数写入到画布中
    }
    for($i=0;$i<200;$i++){                                //for 循环语句生成干扰线
        $randcolor=imagecolorallocate($numimage,rand(200,255),rand(200,255),
rand(200,255));                                           //定义颜色
        imagesetpixel($numimage,rand()%70,rand()%20,$randcolor);  //生成干扰线
    }
    imagepng($numimage);                                  //生成图像
    imagedestroy($numimage);                              //释放资源
?>
```

1.6.4　E-mail 激活技术

为了防止恶意注册，现在有很多网站采用了 E-mail 激活技术。当用户注册成功后，并不是马上就可以使用，而是需要先登录邮箱，通过系统发出的 E-mail 中的超链接进行激活，只有激活后，账号才可以使用。对于找回密码的用户，系统会将新密码发送到邮箱中。这在一定程度上提高了安全性。

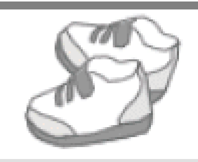
本模块使用 ZendFramework 中的 Zend_Mail 组件完成邮件的发送操作。在本书的资源中没有提供 Zend_Mail 组件，如果要运行这个程序，需要读者自己下载 Zend 组件，并将其复制到本模块的根目录下，即 mr/01 文件夹下。

说明：

在执行邮件的发送时，笔者使用的是搜狐的邮箱，邮箱名称是 mrsoft8888@sohu.com，密码是 mrsoft8888。读者在完成用户注册之后，可以登录这个邮箱完成注册用户的激活操作，否则注册的用户是不能够使用的。

注册用户激活的原理是控制数据库中用户信息表（tb_member）中 active 字段的值。默认注册成功后该字段的值是 0，但是此时该用户是不能够登录网站的，必须通过邮箱激活，将该字段的值更新为 1 之后，此用户才可以使用。激活操作的文件存储于 activation.php 文件中，其关键代码如下：

```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
include_once("conn/conn.php");
if (!empty($_GET['name']) && !is_null($_GET['name'])){                //激活注册用户
    $num=$conne->getRowsNum
("select * from tb_member where name='".$_GET['name']."' and password = '".$_GET['pwd']."'");
    if ($num>0){
        $upnum=$conne->uidRst
("update tb_member set active = 1 where name='".$_GET['name']."' and password =
'".$_GET['pwd']."'");
```






```
                if($upnum > 0){
                    $_SESSION['name'] = $_GET['name'];
                    echo "<script>alert('用户激活成功！');window.location.href='main.php';</script>";
                }else{
                    echo "<script>alert('您已经激活！');window.location.href='main.php';</script>";
                }
            }else{
                echo "<script>alert('用户激活失败！');window.location.href='register.php';</script>";
            }
        }
    ?>
```

1.6.5 应用键盘响应事件验证信息是否合法

新用户注册时，随着信息的输入，系统会实时显示信息的正确性；用户登录时，无须使用鼠标来选择文本框，只要按下 Enter 键，焦点就会自动下移。这些功能都是通过 JavaScript 脚本中的键盘响应事件来实现的。随着 Ajax 技术的流行，JavaScript 已经成为开发人员必须掌握的一项技术。下面就来简单了解一下键盘事件。

用户通过 onkeydown 和 onkeyup 事件来触发响应事件。使用方法和 onclick 事件类似。onkeydown 表示当键盘上的键被按下时触发，onkeyup 和它正好相反，表示当键盘上的键被按下又抬起时触发。在页面中加载事件的方式有多种，这里介绍两种最常用的方式。

（1）将事件直接添加到页面元素中

这种方法最直接、简单。格式如下：

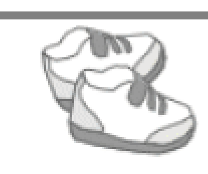
```
<script type="text/javascript">
    …
    function refer(){
        …
    }
</script>
<input type="text" onkeydown="refer()" />
```

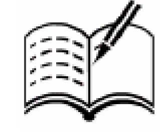
当用户输入完信息后，按任意键，onkeydown 事件即被触发，并调用 refer()函数。

（2）通过 window.onload 加载

当页面被载入时，事件被载入。代码如下：

```
<script>
    window.onload = function(){
        document.getElementById('lgname').onkeydown = function(){
            …
        }`
    }
</script>
…
<input id="lgname" type="text" />
…
```





Note

当用户输入信息时，每输入一个字母，都将触发该事件。在该事件调用的函数中，对用户输入信息进行判断。例如，用户名必须大于等于 2，密码最短 6 位，E-mail 必须合法等。下面看一段验证用户名的代码：

```
//验证用户名
//为 id 等于 regname 的页面元素添加 onkeyup 事件
//用户每按一次键，都会调用一次该函数
$('regname').onkeyup = function (){
        name = $('regname').value;                    //获取输入内容
        cname2 = '';
        if(name.match(/^[a-zA-Z_]*/) == ''){          //判断输入字符是否在有效范围之内
            $('namediv').innerHTML = '<font color=red>必须以字母或下划线开头</font>';
            cname1 = '';
        }else if(name.length < 2){                    //判断输入的字符长度
            $('namediv').innerHTML = '<font color=red>注册名称必须大于等于 2 位</font>';
            cname1 = '';
        }else{
            $('namediv').innerHTML = '<font color=green>注册名称符合标准</font>';
            cname1 = 'yes';
        }
        chkreg();
}
…
```

该段代码的运行结果如图 1.12 所示。

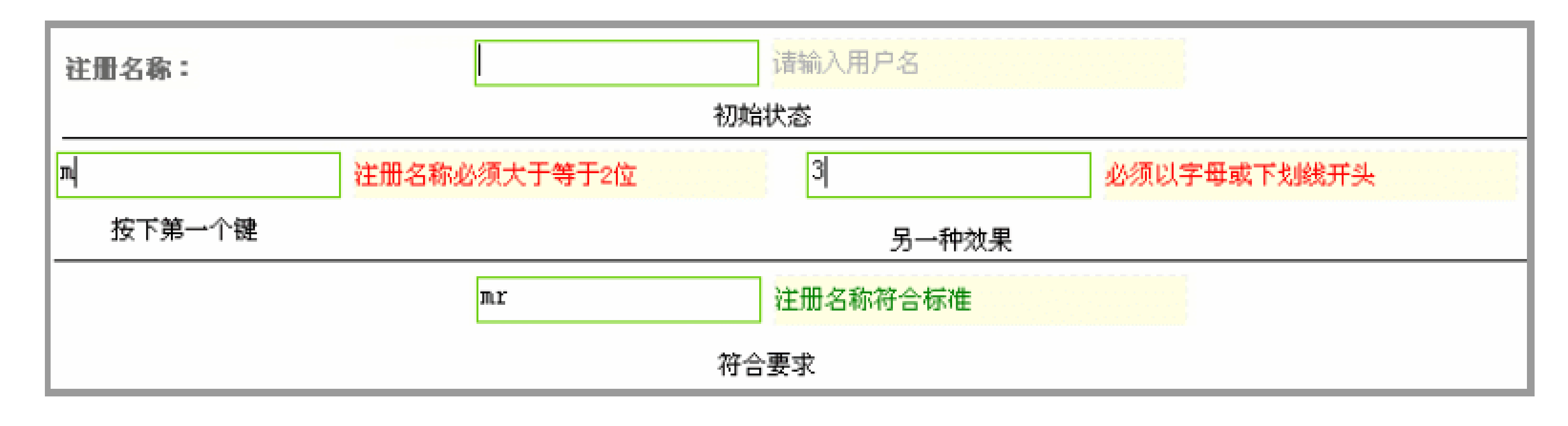
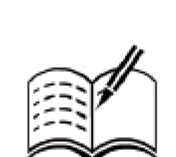


图 1.12　验证信息合法性

使用 onkeydown 事件还可以实现对特定键的控制，包括 Enter 键、Ctrl 键、Alt 键等所有的按键。这是通过在 onkeydown 事件中使用 keyCode 属性来实现的。keyCode 属性能够知道用户按下的是哪个键，例如 Enter 键等于 13，空格键等于 32 等。使用 keyCode 属性的一般格式如下：

```
<script>
    window.onload = function(){
        document.getElementById('lgname') = function(){
            if(event.keyCode == 13){                  //判断用户是否按下了 Enter 键
                …
            }
        }`
    }
</script>
```





在本模块中，实现了焦点自动下移功能。当用户按下 Enter 键时，页面的焦点自动移到下一个文本框中，效果如图 1.13 所示。

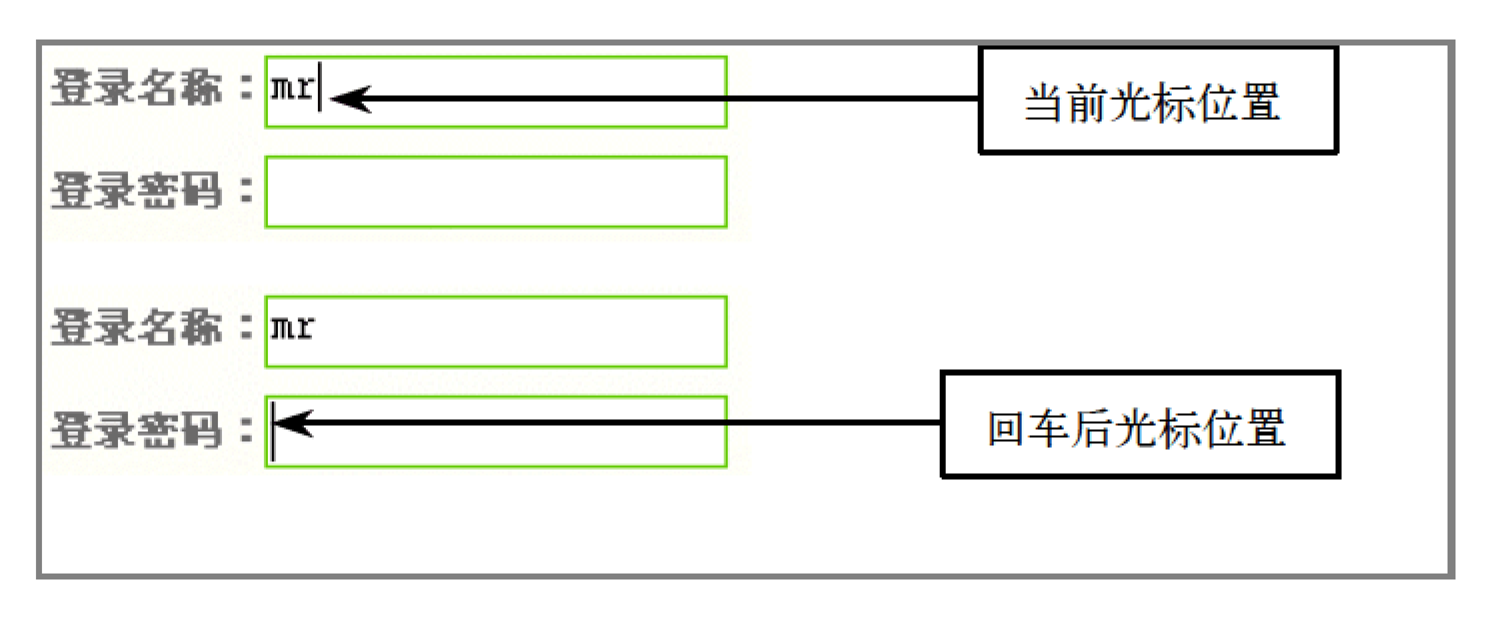


图 1.13　焦点自动下移的运行效果

1.6.6　在 PHP 中操作 Cookie 技术

Cookie 的作用是当用户第一次访问某服务器时，服务器将一些信息保存到客户端计算机内。在以后的一段时间内，当用户再次访问这个服务器时，服务器通过 Cookie 信息能够很快识别该用户。在很长一段时间内，Cookie 都被当作一个不安全的因素，很少再被使用，转而使用 Session。但是，Cookie 的功能确实十分方便，如自动登录、网站统计等，都要比其他的实现方式简单、快捷得多。Cookie 只是一个文本文件，不能访问本地硬盘，无法传播病毒、木马程序。唯一要注意的地方是，Cookie 只能识别计算机，而不会在意谁在使用。

在 PHP 中操作 Cookie，主要用到 setcookie()函数和$_COOKIE 预定义变量。setcookie()函数的语法格式如下：

```
bool setcookie ( string name [, string value [, int expire [, string path [, string domain [, bool secure]]]]] )
```

参数说明如下。

☑ name：设置了 Cookie 的名称。

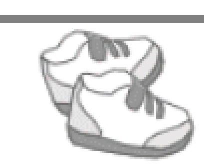
☑ value：Cookie 的值，即参数 name 的值。

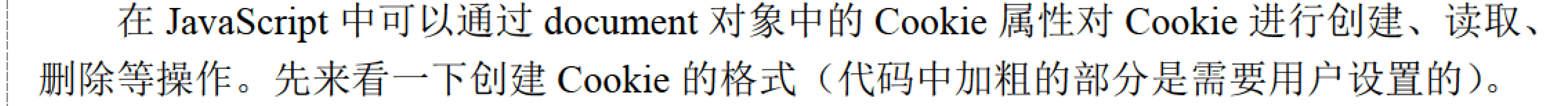
☑ expire：用来设置 Cookie 的过期时间。该参数以时间戳的形式存在。一般设置 Cookie 过期时间的时候，通过“time()+秒数”来实现。例如 time()+60×10，表示 Cookie 将在 10 分钟后失效。如要设置 Cookie 马上失效，可以将时间设为当前日期之前，如 time()−1，那么 Cookie 会立即失效。

☑ path：表示 Cookie 在服务器端的有效范围。如果 path 设为“/”，那么 Cookie 在整个服务器内都有效；如果 Cookie 为/05/，那么 Cookie 只在服务器下的 05 目录有效。

☑ domain：指定了 Cookie 有效的域名范围。以 www.mrbccd.com 为例，如果 domain 设为“.mrbccd.com”，那么 Cookie 在该域的所有子域都有效；如果 domain 为“www.mrbccd.com”，那么 Cookie 只在 www.mrbccd.com 域内有效。

☑ secure：表明是否在 https 传送。如果为 True，表明 Cookie 仅在 HTTPS 连接中被设置；默认是 False。







Note

1.6.7 在 JavaScript 中操作 Cookie 技术

在 JavaScript 中可以通过 document 对象中的 Cookie 属性对 Cookie 进行创建、读取、删除等操作。先来看一下创建 Cookie 的格式（代码中加粗的部分是需要用户设置的）。

```
documen.cookie = "Name=Value; expires=time; path=URL; domain= Domain"
```

其中各个参数的含义和 PHP 中的基本相同，这里就不再介绍了。

在 JavaScript 中读取 Cookie 有一点复杂。在 JavaScript 看来，Cookie 就是一个字符串。例如一个名为 Count、值是 10 的 Cookie，在 JavaScript 中的格式如下。

```
count=10
```

如果有多个 Cookie，那么其格式如下：

```
count=10; name=mr; …
```

而真正需要的值，其实只有 10 而已。所以，要获取 Cookie 的值，就要使用 split()函数将 Cookie 进行拆分。获取 count 值的代码如下：

```
<script>
    if(document.cookie==''){                        //Cookie 等于空时
        document.write('no cookie');
    }else{
        cookiearr = document.cookie.split('; ');    //以分号为分隔符，对 Cookie 进行分割
        leng = cookiearr.length;                    //获取数组长度
        for(i=0; i<leng; i++){
            if(cookiearr[i].split('=')[0] == 'count'){ //判断哪一个 Cookie 等于 count
                document.write(cookiearr[i].split('=')[1]);    //输出 count 的值
            }
        }
    }
</script>
```

1.6.8 用户自动登录技术

自动登录的原理是：当用户打开登录页面时，登录页面首先判断用户客户端机器中的 Cookie 值。如果该值不存在，或者 Cookie 已失效，将显示登录表单；如果 Cookie 值存在，则直接使用 Cookie 中所保存的用户名进行登录。

在本模块中，用户首先打开的是 index.php 页，该页判断 Cookie 值，如果没有，则跳转到 login.php 进行登录；如果 Cookie 有值，则将 Cookie 中的用户名保存到 Session，直接进入主页。实现代码如下：

```
<?php
    session_start();                                            //开启 Session 支持
    header('Content-Type:text/html;charset=gb2312');            //设置页面编码
    if(!empty($_COOKIE['name']) and !is_null($_COOKIE['name'])){ //判断 Cookie 是否
为空
```



```
        $_SESSION['name'] = $_COOKIE['name'];           //将 Cookie 保存到 Session 中
        header('location:http://localhost/model/05/01/main.php'); //跳转到 main.php 页
    }else{                                              //Cookie 为空，说明没有登录
        header('location:http://localhost/model/05/01/login.php');//跳转到 login.php 页
    }
?>
```



这里$_COOKIE['name']的值是在登录成功时设置的，相关代码如下。

```
<?php
    …                                                   //如果登录成功
    setcookie('name',$name,time()+60*10);               //保存$name 的同时设置时间
    …
?>
```

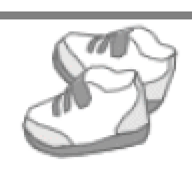
第 2 章

分页显示模块

（自学视频、源程序：配套资源\mr\2\）

在网站中，当面对数据库中的大量数据要输出时，在头脑中闪过的第一个想法就是应用什么分页技术输出数据库中的数据才更加合理、简单、快捷。分页功能是每个网站都必不可少的。在本模块中，将介绍几种分页的实现方法，希望对读者以后程序的开发有所帮助。通过本章的学习，读者能够学到以下内容。

- ⏭ 自定义函数替换超长文本中的特殊字符
- ⏭ 自定义函数控制超长文本的输出
- ⏭ 通过 PDO 操作 MySQL 数据库
- ⏭ PHP 超长文本分页
- ⏭ Ajax 无刷新分页
- ⏭ 分页类分页
- ⏭ PHP 上下分页
- ⏭ PHP 跳转分页



Note

2.1 分页显示概述

2.1.1 模块概述

分页显示模块是网站开发中最常用的一个典型模块，其最大用途就是将海量数据或者超长文本数据分页输出，这样一来不仅使页面的整体效果更加紧凑，而且能给用户带来最好的体验。

2.1.2 功能结构

在本模块中，整体上分为两部分内容，一部分是面向对象的分页，另一部分是面向过程的分页。在面向对象分页中，通过 Smarty 模板完成网页的动静分离，通过 PDO 操作 MySQL 数据库，并且将 Smarty 模板的配置、PDO 连接操作数据库以及分页方法都封装到类中，通过类中方法的调用完成各种操作；在面向过程分页中，完成超长文本的分页、Ajax 无刷新分页、跳转分页和上下分页。本模块的系统功能结构如图 2.1 所示。

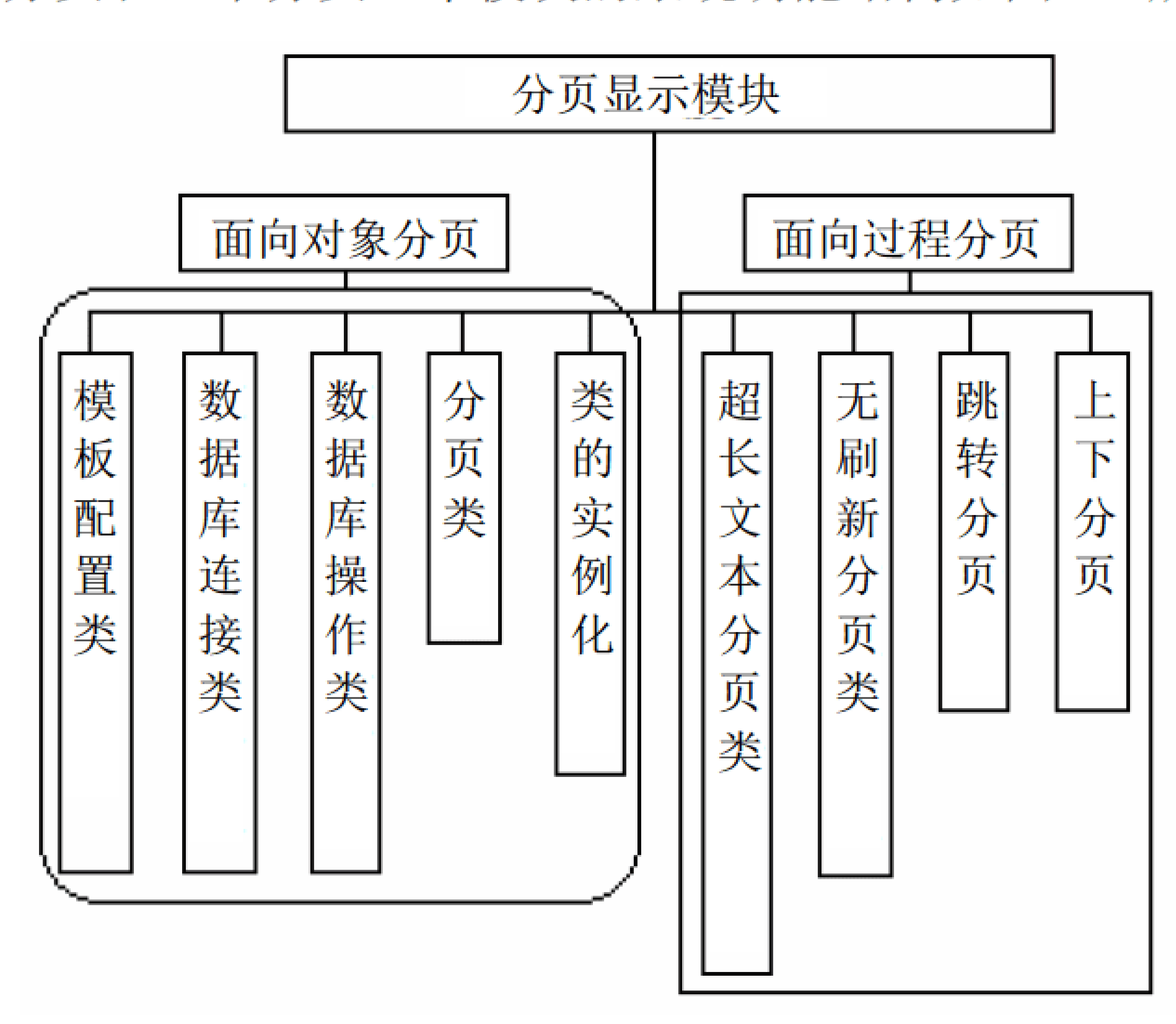


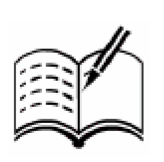
图 2.1　分页显示模块功能结构图

2.1.3 程序预览

在分页显示模块中展示了 6 种不同的分页方法：超长文本的分页显示、Ajax 无刷新分页、标准分页、跳转分页、上下分页和通过分页类分页。

- ☑ 超长文本的分页显示：实现对文本文件中数据的分页显示，主要应用在注册须知、公司简介、版权说明等文本类的文件输出中，如图 2.2 所示。
- ☑ Ajax 无刷新分页：实现数据库中数据的无刷新显示。这种分页方法主要应用在一些支持视频播放的网站中，通过 Ajax 无刷新进行分页，不会影响到视频文件的





Note

播放，如图 2.3 所示。



图 2.2　超长文本的分页显示



图 2.3　Ajax 无刷新分页的应用

☑ 上下分页：针对当前的页面，实现上一页和下一页的跳转，也属于一种常用的分页方法，如图 2.4 所示。



图 2.4　上下分页



☑ 通过分页类分页：将一种定义好的分页方法封装到一个类中，当需要进行分页操作时，可以直接调用这个类文件，并设置相应的参数，即可实现分页的操作，无须再重新编写分页方法，如图 2.5 所示。这是一种比较实用的分页方法。



图 2.5　通过分页类分页

2.2　面向对象分页

通过分页类实现分页功能是一种非常好的方法。一个分页类创建成功后，就可以直接调用类文件实现分页功能，而无须再重新编译。在分页类模块中，采用 Smarty 模板完成网页的动静分离，通过 PDO 完成对数据库的连接和操作。通过分页类实现分页的效果如图 2.6 所示。



图 2.6　通过分页类分页的运行效果

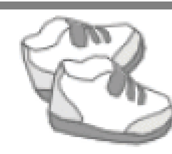
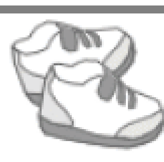
2.2.1　Smarty 模板的安装和配置

首先，将 Smarty 模板的压缩包解压，复制 libs 文件夹到服务器 mr\2\01\system 目录下。

接下来，配置 Smarty 模板。在确定 Smarty 模板文件的位置之后，在 smarty 目录下新建 4 个目录——templates、templates_c、configs 和 cache，用于存储不同的文件。

最后，封装 Smarty 模板的配置类，完成 Smarty 模板的配置操作。该类存储于 2\01\system\system.smarty.inc.php 文件中，其代码如下：

```
<?php
```


Note

```
require("libs/Smarty.class.php");                          //包含模板文件
class SmartyProject extends   Smarty{                       //定义类，继承模板类
        function SmartyProject(){                           //定义方法
                $this->template_dir = "./";                 //指定模板文件存储位置
                $this->compile_dir = "./system/templates_c/"; //指定编译文件存储位置
                $this->config_dir = "./system/configs/";    //指定配置文件存储位置
        $this->cache_dir = "./system/cache/";               //指定缓存文件存储位置
        }
}
?>
```

2.2.2　加载 PDO 数据库抽象层

在本模块中，通过 PDO 数据库抽象层连接和操作 MySQL 数据库。PDO 是与 PHP 5.1 一起发行的，默认包含在 PHP 5.1 中。由于 PDO 需要 PHP 5 核心面向对象特性的支持，因此其无法在 PHP 5.0 之前的版本中使用。

默认情况下，PDO 在 PHP 5.2 中为开启状态，但是要启用对某个数据库驱动程序的支持，仍需要进行相应的配置操作。

在 Linux 环境下，要使用 MySQL 数据库，可以在 configure 命令中添加如下选项。

```
--with-pdo-mysql=/path/to/mysql/installation
```

在 Windows 环境下，PDO 在 php.ini 文件中进行配置，如图 2.7 所示。

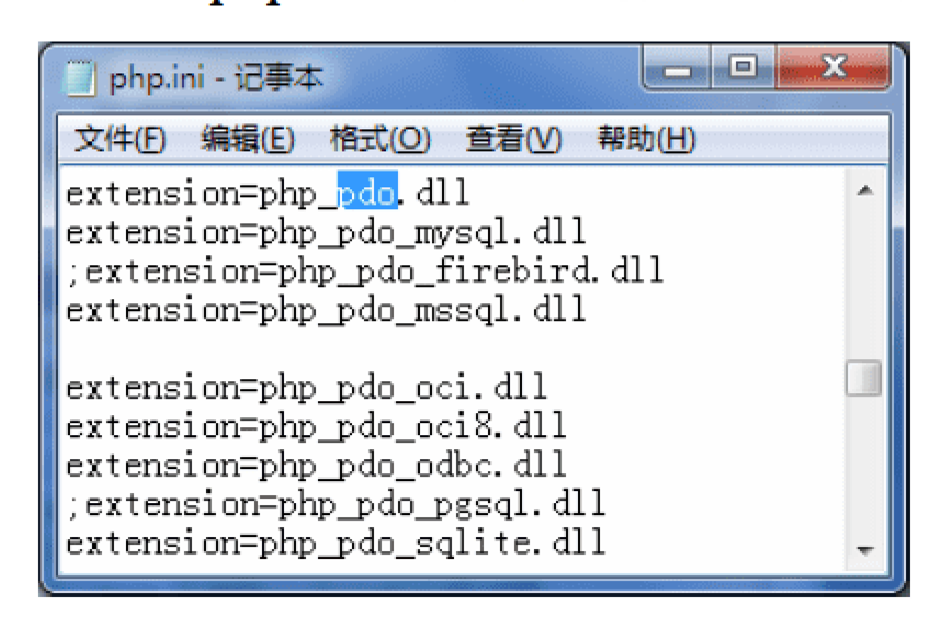


图 2.7　在 Windows 环境下配置 PDO

要启用 PDO，首先必须加载“extension=php_pdo.dll”。要想让其支持某个具体的数据库，那么还要加载对应的数据库选项。例如，要支持 MySQL 数据库，则需要加载“extension=php_pdo_mysql.dll”选项。

注意：

在完成数据库的加载后，要保存 php.ini 文件，并且重新启动 Apache 服务器，修改才能够生效。有关 PDO 连接和操作 MySQL 数据库的方法已经在关键技术中作了详细讲解，这里不再赘述。

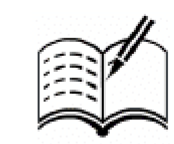
2.2.3　分页类模块的页面设计

由于本模块的开发使用的是 Smarty 模板技术，因此模块的页面设计和程序的开发是



完全分离的，模块的页面设计是在 index.tpl 文件中完成的，而程序的开发则是在 index.php 文件中完成。

分页类模块的页面设计主要就是将程序开发中模板变量中的数据进行输出，包括图书名称、图书介绍、出版时间和作者，以及分页功能的实现。关键代码如下：

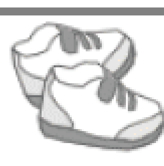
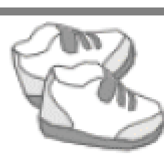



```
{if $isbbs=="T"}          <!--首先通过模板变量$isbbs 判断是否有数据-->
<!--通过 section 内建函数，循环输出数组模板变量$arraybbs 中的图书数据-->
{php} $i=1;   {/php}
{section name=bbsid loop=$arraybbs}
<tr>
        <td height="22" class="STYLE4" {php} if($i%2==0){ echo "bgcolor='#F5F5F5'";} {/php}> 
             <a href="index.php?id={$arraybbs[bbsid].tb_book_id}" target="_blank">
{unhtml content=$arraybbs[bbsid]. tb_book_name}</a>
        </td>
        <td class="STYLE4" {php} if($i%2==0){ echo "bgcolor='#F5F5F5'";} {/php}>
<div align="center">{$arraybbs [bbsid].tb_book_synopsis}</div></td>
        <td height="22" class="STYLE4"  {php} if($i%2==0){ echo"bgcolor='#F5F5F5'";} {/php}>
<div align="center"> {$arraybbs[bbsid].tb_print_date}</div></td>
        <td height="22" class="STYLE4" {php} if($i%2==0){ echo "bgcolor='#F5F5F5'";} {/php}>
<div align="center"> {$arraybbs[bbsid].tb_book_author}</div></td>
</tr>
{php} $i++;   {/php}
{/section}
<table width="100%" height="22" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr><!--输出分页功能的信息-->
             <td class="STYLE4"> {$showpage}</td>
          </tr>
</table>
{/if}
<!--判断当没有数据时，则输出如下内容-->
{if $isbbstell=="F" && $isbbs=="F"}
<table width="96%" height="30" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#58BAE9">
        <tr>
       <td bgcolor="#FFFFFF"><div align="center">
<font color="#FF0000">{ldelim}该分类暂无论坛公告或帖子{rdelim}</font></div></td>
        </tr>
</table>
{/if}
```

2.2.4 分页类模块的程序开发

分页类模块的程序开发是在 index.php 文件中进行的，是一个纯粹的 PHP 代码页。在该文件中调用类文件，实现与数据库的连接，读取数据库中的数据；调用分页类，实现数据的分页显示，并且定义模板变量，用于在模板页中输出数据。关键代码如下：

```
<?php
header("Content-Type:text/html;charset=gb2312");
```

```
require_once("system/system.inc.php");          //调用指定的文件
$arraybbstell=$admindb->ExecSQL("select * from tb_mr_book",$conn); //执行 select 查询语句
if(!$arraybbstell){
    $smarty->assign("isbbstell","F");//判断如果执行失败则输出模板变量 isbbstell 的值为 F
}else{
    $smarty->assign("isbbstell","T");//判断如果执行成功，则输出模板变量 isbbstell 的值为 T
    $smarty->assign("arraybbstell",$arraybbstell);        //定义模板变量 arraybbstell，输出数据库中数据
}
if(isset($_GET["page"])){
    $page=$_GET["page"];
}else{
    $page=1;
}
$arraybbs=$seppage->ShowData("select * from tb_mr_book",$conn,1,$page);   //调用分页类，实现分页功能
if(!$arraybbs){
    $smarty->assign("isbbs","F");
}else{
    $smarty->assign("isbbs","T");
$smarty->assign("showpage",$seppage->ShowPage("","","","","")); //定义输出分页数据的模板变量 showpage
    $smarty->assign("arraybbs",$arraybbs);
}
$smarty->display("index.tpl");                                  //指定连接的模板页
?>
```

至此，分页类模块的讲解完毕，程序的完整代码参见配套资源中的相应内容。

2.3 超长文本分页

2.3.1 超长文本分页功能概述

利用 PHP 的超长文本分页显示功能，可将指定文本文件中的数据进行分页输出，效果如图 2.8 所示。



图 2.8 PHP 超长文本分页显示的运行效果



2.3.2　超长文本分页功能实现过程

PHP 超长文本的分页显示主要应用的是自定义函数 unhtml()和 msubstr()，有关这两个自定义函数的创建参见 2.8.1、2.8.2 节，这里不再赘述。

PHP 超长文本分页显示的操作是在 02\mr_copyright.php 文件中完成的。首先，判断变量的值是否为空，并调用指定的包含文件，获取自定义函数。其代码如下：

```
if(isset($_POST['page'])){
        $page=$_POST['page'];
}else if(isset($_GET['page'])){
        $page=$_GET['page'];
}else{
        $page=1;
}
include("function.php");                    //包含自定义函数
```

然后，通过 file_get_contents()函数获取 file.txt 文本文件中的数据，返回到一个字符串中；通过 unhtml()函数替换字符串中的特殊字符；通过 strlen()函数获取字符串的长度；计算将文本文件中的数据分成几页进行输出；通过 msubstr()函数控制数据的输出；调用 substr()函数获取当前页输出的内容。代码如下：

```
<?php
    if($page){
        $counter=file_get_contents("file/file.txt");//读取文本文件中的数据，返回一个字符串
        $length=strlen(unhtml($counter));           //计算字符串的长度，替换特殊字符
        $page_count=ceil($length/1050);             //对字符串进行分页处理
        $c=msubstr($counter,0,($page-1)*1050);      //通过自定义函数获取上一页中的数据
        $c1=msubstr($counter,0,$page*1050);         //通过自定义函数获取当前页中的数据
            echo substr($c1,strlen($c),strlen($c1)-strlen($c)); //获取到当前页中要输出的数据
    }
    ?>
```

最后，创建分页超链接，实现首页、上一页、下一页和尾页之间的跳转。代码如下：

```
<td width="42%" align="center" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">   页 次 ： <?php echo $page;?> / <?php echo $page_count;?>页</span></td>
    <td width="58%" height="28" align="left" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">  分页：
    <?php
    if($page!=1){
            echo "<a href=index.php?lmbs=明日版权声明&page=1>首页</a> ";
            echo "<a href=index.php?lmbs=明日版权声明&page=".($page-1).">上一页</a> ";
     }
     if($page<$page_count){
            echo "<a href=index.php?lmbs=明日版权声明&page=".($page+1).">下一页</a> ";
            echo "<a href=index.php?lmbs=明日版权声明&page=".$page_count.">尾页
```


Note

```
</a>";
        }
        ?>
      </span> </td>
```

至此，PHP 超长文本的分页显示介绍完毕。

2.4　Ajax 无刷新分页

2.4.1　Ajax 无刷新分页功能概述

Ajax 无刷新分页的强大功能体现最为明显的地方就是在播客中。在播客中，在播放视频文件的同时，如果使用的不是无刷新分页，那么当执行翻页的操作后，视频文件将被重新打开；如果使用的是无刷新分页，就不会出现这种情况。

为了更好地理解 Ajax 无刷新分页的强大功能，这里创建了一个视频文件，并且应用 Ajax 无刷新分页输出文本文件中的内容，验证无刷新分页的操作是否对视频文件的播放有影响。运行效果如图 2.9 所示。



图 2.9　Ajax 无刷新分页的应用

2.4.2　Ajax 无刷新分页功能实现过程

Ajax 无刷新分页由 mr_synopsis.php、mr_synopsis_ok.php 和 discuss_js.js 3 个文件组成。

在 mr_synopsis.php 文件中，首先创建一个 div 标签，设置其 id="synopsis"；然后在 div 标签中编写超长文本分页输出的代码，并且在超长文本分页的超链接中应用 onclick 事件，调用自定义函数 artpagination()，传递的参数为另外一个实现分页功能的文件 mr_synopsis_ok.php。其关键代码如下：

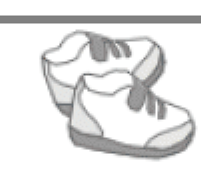
```
<div id="synopsis"> <!--创建 div 标签，用于获取 JS 文件中返回的分页结果-->
<table width="425" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td height="20"> </td>
```



Note

```
            </tr>
            <tr>
              <td>
<?php
//读取超长文本中的数据，实现超长文本中数据的分页显示
  if($page){
        $counter=file_get_contents("file/mr_synopsis.txt");
        $length=strlen($counter);
        $page_count=ceil($length/950);
        $c=msubstr($counter,0,($page-1)*950);
        $c1=msubstr($counter,0,$page*950);
            echo substr($c1,strlen($c),strlen($c1)-strlen($c));
  }
?>      </td>
            </tr>
            <tr>
              <td>
<table border="0" align="center" cellpadding="0" cellspacing="0">
  <tr><!--设置超长文本分页显示的超链接-->
    <td width="42%" align="center" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">   页 次 : <?php echo $page;?> / <?php echo $page_count;?>页</span></td>
    <td width="58%" height="28" align="left" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">  分页：
                        <?php
                          if($page!=1){
?>
<!--调用 artpagination 函数，实现无刷新的分页输出-->
<a href="#" onclick='return artpagination("mr_synopsis_ok.php?page=1")'>首页</a> 
<a href="#" onclick='return artpagination("mr_synopsis_ok.php?&page=<?php echo $page-1;?>")'>上一页</a> 
                              <?php  }
                               if($page<$page_count){
?>
<a href="#" onclick='return artpagination("mr_synopsis_ok.php?page=<?php echo $page+1;?>")'>下一页</a> 
 <a href="#" onclick='return artpagination("mr_synopsis_ok.php?page=<?php echo $page_count;?>")'>尾页</a>
<?php
                   }
                          ?>
    </span> </td>
  </tr>
</table></td>
            </tr>
    </table>
</div>
```

最后，通过 js 文件中的自定义函数 artpagination()将 mr_synopsis_ok.php 文件中的分页



结果返回到 div 标签 synopsis 中，实现无刷新分页的功能。js 文件中的关键代码如下：

```
//使用 XMLHttpRequest 对象创建异步 HTTP 请求
function artpagination(url){                //创建自定义函数，获取传递的参数
    xmlHttp.open('get',url,true);           //通过 get 方法，执行另外一个实现分页功能的文件
    xmlHttp.onreadystatechange = function(){
        if(xmlHttp.readystate == 4 && xmlHttp.status == 200){ //将结果返回到 div 标签 synopsis 中
            document.getElementById("synopsis").innerHTML = xmlHttp.responseText;
        }
    }
    xmlHttp.send(null);
}
```

说明：

在 mr_synopsis_ok.php 文件中完成的分页功能与 mr_synopsis.php 文件中是相同的，代码不再重复。

2.5 标 准 分 页

2.5.1 PHP 标准分页功能概述

利用 PHP 的标准分页功能，可在当前页面中显示指定的记录；也可以 10 页为一个单位，实现上 10 页和下 10 页的跳转；此外，还可实现当前页中显示的 10 个超链接的任意跳转。其运行效果如图 2.10 所示。



图 2.10　PHP 标准分页运行效果

2.5.2 PHP 标准分页功能实现过程

在 PHP 标准分页中，每页显示 5 条数据。分页跳转可以分为 3 个部分：（1）根据查询结果，通过 while 语句在当前页面中显示 5 条记录信息；（2）以 10 页为一个单位，实现上



10 页和下 10 页的跳转；（3）在当前页中，在显示的 10 个超链接中任意跳转。PHP 标准分页是在 mr_book.php 文件中完成的，其具体操作步骤如下。

（1）连接数据库，定义分页中要使用的变量，包括数据库的连接文件 conn.php、当前页值的变量$page 和当前 10 页值的变量$link_type。关键代码如下：

```
<?php
 include_once("conn/conn.php");
if(isset($_POST['page'])){
        $page=$_POST['page'];
}else if(isset($_GET['page'])){
        $page=$_GET['page'];
}else{
        $page=1;
}
if (isset($_GET['link_type'])) {
        $link_type=$_GET['link_type'];
}else{
        $link_type=0;
}
if (isset($_GET['vv'])) {
        $vv=$_GET['vv'];
}else{
        $vv=0;
}
include("function.php");
?>
```

说明：

因为本模块应用 switch 语句完成网页的框架操作，所有功能都包含在 index.php 文件中，所以将上述的这段代码存储在 index.php 文件中，作为公共内容。

（2）在 mr_book.php 文件中，从数据库中读取数据，统计出总的记录数，设置每页显示的记录数，计算出共有几页，并通过 while 语句循环输出数据库中的数据。代码如下：

```
<?php
if($page){                                          //判断当前页变量的值是否存在
        $page_size=5;                               //定义每页输出的记录数
       $query="select * from tb_forum_send ";      //编写查询语句，应用 count 统计总的记录数
        $result=$conn->query($query);               //执行查询语句
        $res=$result->fetchAll(PDO::FETCH_ASSOC);
        $message_count=count($res);
       $page_count=ceil($message_count/$page_size);//计算出总共有几页
       $offset=($page-1)*$page_size;               //输出上一页结束的记录数
        $query_2=$conn->query("select * from tb_forum_send limit $offset, $page_size");
        foreach($query_2 as $value){
?>
   <tr>
```



```
		<td width="35%" align="center" bgcolor="#FFFFFF">
			<a href="index_ok.php?send_id=<?php echo $value['tb_send_id'];?>" target="_blank">
	<?php echo $value['tb_send_subject'];?></a>
		</td>
		<td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_date'];?></td>
		<td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_user'];?></td>
		<td width="10%" align="center" bgcolor="#FFFFFF">
		<?php
				$query_s=$conn->query("select * from tb_forum_restore ");
				echo count($query_s);
			?>
		</td>
	</tr>
	<?php
		}
	}
	?>
```

（3）创建分页超链接，实现分页中以 10 页为单位的跳转，以及当前 10 页中不同页面之间的跳转。关键代码如下：

```
	<table width="800" height="30" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#D0D0D0">
		<tr>
		<td height="22" align="center" bgcolor="#FFFFFF">
	<span class="STYLE2">帖子统计：<?php echo $message_count;?> 条  </span></td>
			<td align="center" bgcolor="#FFFFFF">每页显示：<?php echo $page_size; ?>条</td>
			<td align="center" bgcolor="#FFFFFF">共有：<?php echo $page_count;?>页</td>
			<td align="center" bgcolor="#FFFFFF"><span class="STYLE2"> 分页：
	<?php
	$next=$link_type*10; //以 10 页为一个单位
	$n=$link_type-1;	//上一个 10 页的值
	$m=$link_type+1;	//下一个 10 页的值
	$prev_page=$page-10;//当前页的上 10 页的值
	$mm="";
	if($link_type==0){	//判断变量的值是否等于 0，如果等于 0 则输出首页图标，无超链接
		echo "<img src=\"images/02.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"首页\">";
	}else{	//否则，为首页图标设置超链接，连接到首页，并为上 10 页的图标设置超链接
		echo "<a href='index.php?lmbs=明日论坛&vv=0&link_type=0&page=1'>
	<img src=\"images/02.jpg\" width=\"8\" border=\"0\" height=\"9\" border=\"0\" title=\"首页\"></a> ";
		$ccc=$vv-10;		//定义变量，变量$vv 是当前 10 页的起始位置，$ccc 是上一个 10 页的起始位置
		echo "<a href='index.php?lmbs=明日论坛&vv=$ccc&link_type=$n&page=$prev_page'>
	<img src=\"images/01.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"上十页\"></a>";
	}
	?>
	<?php
	//通过 for 循环语句，输出当前页中的 10 个超链接
```




```
for($j=1;$j<=$page_count;$j++) {                //根据统计的总的记录数循环输出
	$pnext=$next+$j;                            //定义当前页中输出的页码值
	if($mm==10){                                //判断当$mm 的值等于 10 时跳出循环
		break;
	}
	if($mm>$page_count){                        //判断当$mm 的值大于总记录时跳出循环
		break;
	}
	if($page_count-$vv<10){
	if($mm>=$page_count-$vv){                   //判断当$mm 的值大于等于总记录数减去当前 10 页的起始值时，跳出循环
			break;
		}
	}
	echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$link_type &page =$pnext'>";
	if($page==$pnext){                          //为当前单击的页码描红
		echo "<span class='STYLE1'> $pnext </span>";
	}else{
		echo " $pnext ";                        //输出当前 10 页超链接中的数据
	}
	echo "</a>";
	$mm=$mm+1;
}
?>
</span><span class="STYLE2">
<?php
$vv=$vv+$mm;                 //定义变量的值，定义下 10 页的起始位置
if($page_count-$vv<=0){  //判断当总记录数小于等于下 10 页的起始位置时，输出尾页的图标
	echo "<img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"尾页\">";
}else{                         //否则输出下 10 页的超链接
		echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$m&page= $pnext'>
<img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"下十页\"></a>";
}
if($message_count==0){      //判断当总记录数等于 0 时，输出没有记录
	echo "没有记录!";
}
?>
		</span>  </td>
	</tr>
</table>
```

至此，有关 PHP 标准分页的功能介绍完毕，详细代码参见配套资源中的相应内容。

2.6　跳 转 分 页

2.6.1　PHP 跳转分页功能概述

PHP 跳转分页的特点是能够实现从当前页面到其他指定页面的跳转，运行效果如

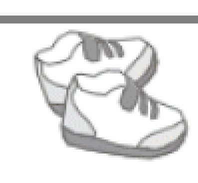


图 2.11 所示。多数情况下它与标准分页功能结合使用，以增加分页操作的灵活性。



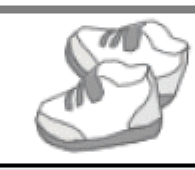
图 2.11　PHP 跳转分页的运行效果

2.6.2　PHP 跳转分页功能的实现过程

在 PHP 跳转分页中，每页显示 5 条数据。分页跳转可以分为 3 个部分：（1）通过文本框实现指定页面的跳转功能；（2）以 10 页为一个单位，实现上 10 页和下 10 页的跳转；（3）在当前页中，在显示的 10 个超链接中任意跳转。PHP 跳转分页是在 mr_forum.php 文件中完成的，其具体操作步骤如下。

（1）连接数据库，定义分页中要使用的变量，包括数据库的连接文件 conn.php、当前页值的变量$page 和当前 10 页值的变量$link_type。关键代码如下：

```
<?php
 include_once("conn/conn.php");
if(isset($_POST['page'])){
        $page=$_POST['page'];
}else if(isset($_GET['page'])){
        $page=$_GET['page'];
}else{
        $page=1;
}
if (isset($_GET['link_type'])) {
        $link_type=$_GET['link_type'];
}else{
        $link_type=0;
}
if (isset($_GET['vv'])) {
        $vv=$_GET['vv'];
}else{
        $vv=0;
}
include("function.php");
?>
```





说明：

因为本模块应用 switch 语句完成网页的框架操作，所有功能都包含在 index.php 文件中，所以将上述的这段代码存储在 index.php 文件中，作为公共内容。

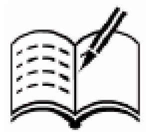
（2）在 mr_forum.php 文件中，从数据库中读取数据，统计出总的记录数，设置每页显示的记录数，计算出共有几页，并通过 while 语句循环输出数据库中的数据。代码如下：

```
<?php
if($page){                                    //判断当前页变量的值是否存在
    $page_size=5;                             //定义每页输出的记录数
    $query="select * from tb_forum_send "; //编写查询语句，应用 count 统计总的记录数
    $result=$conn->query($query);             //执行查询语句
    $res=$result->fetchAll(PDO::FETCH_ASSOC);
    $message_count=count($res);
    $page_count=ceil($message_count/$page_size);       //计算出总共有几页
    $offset=($page-1)*$page_size;             //输出上一页结束的记录数
    $query_2=$conn->query("select * from tb_forum_send limit $offset, $page_size");
    foreach($query_2 as $value){
?>
  <tr>
    <td width="35%" align="center" bgcolor="#FFFFFF">
        <a href="index_ok.php?send_id=<?php echo $value['tb_send_id'];?>" target="_blank">
<?php echo $value['tb_send_subject'];?></a>
     </td>
    <td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_date'];?></td>
    <td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_user'];?></td>
     <td width="10%" align="center" bgcolor="#FFFFFF">
     <?php
              $query_s=$conn->query("select * from tb_forum_restore ");
              echo count($query_s);
         ?>
     </td>
  </tr>
<?php
     }
}
?>
```

（3）创建一个表单，用于提交跳转的页码值，并且通过 JavaScript 脚本判断提交的值是否有效。代码如下：

```
<form name="form1" method="post" action="index.php?lmbs=明日论坛" onSubmit="return chk()">
     <input type="submit" name="Submit" value="跳转"> 
     <input name="page" type="text" size="3">
     <input type="hidden" name="pages" value="<?php echo $page_count;?>">
</form>
```





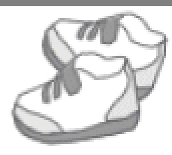
Note

判断提交的页码值是否有效是通过自定义函数 chk()完成的，关键代码如下。

```
<script language="javascript">
  function chk(){
    if(form1.page.value<=0 || parseInt(form1.page.value)>parseInt(form1.pages.value)){
       alert("您输入的页码无效!!");
       form1.page.focus();
       return(false);
     }
     return(true);
  }
</script>
```

（4）编写 PHP 语句，实现分页中以 10 页为单位的跳转，以及当前 10 页中不同页面之间的跳转。关键代码如下：

```
<td align="center" bgcolor="#FFFFFF"><span class="STYLE2"> 分页：
<?php
$next=$link_type*10;        //以 10 页为一个单位
$n=$link_type-1;            //上一个 10 页的值
$m=$link_type+1;            //下一个 10 页的值
$prev_page=$page-10;        //当前页的上 10 页的值
$mm="";
if($link_type==0){     //判断变量的值是否等于 0，如果等于 0 则输出首页图标，无超链接
      echo "<img src=\"images/02.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"首页\">";
}else{                 //否则，为首页图标设置超链接，连接到首页，并为上 10 页的图标设置超链接
      echo "<a href='index.php?lmbs=明日论坛&vv=0&link_type=0&page=1'>
<img src=\"images/02.jpg\" width=\"8\" border=\"0\" height=\"9\" border=\"0\" title=\"首页\"></a> ";
      $ccc=$vv-10;     //定义变量，变量$vv 是当前 10 页的起始位置，$ccc 是上一个 10 页的起始位置
     echo "<a href='index.php?lmbs=明日论坛&vv=$ccc&link_type=$n&page=$prev_page'>
<img src=\"images/01.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"上十页\"></a>";
}
?>
<?php
//通过 for 循环语句，输出当前页中的 10 个超链接
for($j=1;$j<=$page_count;$j++) {         //根据统计的总的记录数循环输出
      $pnext=$next+$j;                   //定义当前页中输出的页码值
      if($mm==10){                       //判断当$mm 的值等于 10 时跳出循环
            break;
      }
      if($mm>$page_count){               //判断当$mm 的值大于总记录数时跳出循环
            break;
      }
      if($page_count-$vv<10){
      if($mm>=$page_count-$vv){          //判断当$mm 的值大于等于总记录数减去当前 10 页的起始值时，跳出循环
                  break;
            }
```



```
            }
            echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$link_type&page =$pnext'>";
            if($page==$pnext){                      //为当前单击的页码描红
                echo "<span class='STYLE1'> $pnext </span>";
            }else{
                echo " $pnext ";                    //输出当前 10 页超链接中的数据
            }
            echo "</a>";
           $mm=$mm+1;
    }
    ?>
    </span><span class="STYLE2">
    <?php
    $vv=$vv+$mm;                                    //定义变量的值，定义下 10 页的起始位置
    if($page_count-$vv<=0){ //判断当总记录数小于等于下 10 页的起始位置时，输出尾页的图标
            echo "<img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"尾页\">";
    }else{                                          //否则输出下 10 页的超链接
                echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$m&page=$pnext'>
    <img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"下十页\"></a>";
    }
    if($message_count==0){                          //判断当总记录数等于 0 时，输出没有记录
            echo "没有记录!";
    }
    ?>
    </span>  </td>
```

至此，有关 PHP 跳转分页的功能介绍完毕，详细代码参见配套资源中的相应内容。

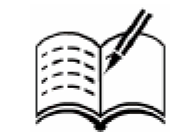
2.7 上 下 分 页

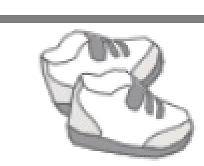
2.7.1 PHP 上下分页功能概述

PHP 上下分页功能以当前页码为基础，实现上一页和下一页的跳转，并且输出上一页中最后一条数据和下一页中第一条数据的内容。运行效果如图 2.12 所示。



图 2.12　PHP 上下分页的运行效果





2.7.2 PHP 上下分页功能的实现过程



Note

在 PHP 上下分页中，每页显示 2 条数据。分页跳转可以分成两个部分：（1）上下翻页的跳转；（2）以 10 页为一个单位，实现上 10 页和下 10 页的跳转，以及在当前页显示的 10 个超链接中的任意跳转。有关第二部分的内容已经在 2.6.2 节中作了详细介绍，这里不再赘述。

下面主要介绍上下翻页的跳转，以及输出上一页的最后一条数据和下一页的第一条数据的方法。此外，还将讲解分栏功能的实现方法。其操作是在 mr_dictionary.php 文件中完成的。

在输出数据库中的数据时，应用的是分栏方法，创建一个 1 行 2 列的表格来输出数据库中的数据。实现的原理如下：

（1）在 while 语句之外，定义一个变量$number=0。

（2）在 while 语句中，通过 if 语句判断取模（$number % 2）的值是否等于 0，如果等于 0，则是行的开始（执行<tr>）；如果不等于 0，则是行的结束（执行<\tv>）。

（3）在 while 语句之外，执行$number++。

其关键代码如下：

```
<?php
        $number=0;                                              //定义变量
        //省略了部分 PHP 代码
        $query_2=mysql_query("select * from tb_mr_bccd limit $offset, $page_size");
        while($myrow_2=mysql_fetch_array($query_2)){  //循环输出
                if (( $number % 2 ) == 0){                      //判断取模的值是否等于 0
?>
                <tr>                                    <!--在取模的值等于 0 时执行此操作-->
<?php           }       ?>
                        <td colspan="2" align="center">
                            <!--输出数据库中数据-->
                        </td>
<?php
        if ( ( $number %2 ) != 0){                              //判断当取模的值不等于 0 时
?>
                </tr>                           <!--当取模的值不等于 0 时执行此操作-->
<?php }                                                         //结束 while 循环
 $number++;                                                     //变量值增加
?>
```

上一页、下一页跳转功能的实现，关键就是对当前页值 page 的判断。设置上一页的超链接和获取上一页中的最后一条数据，关键代码如下：

```
<td width="205" align="center" bgcolor="#FFFFFF">上一部：<a href="index.php?lmbs=明日编程词典&page=<?php
if($page=="" or $page<=1){
        echo 1;
}else{
        echo ceil(($page*2-2)/2);
}
```



```
?>">
<?php
$page_3=($page-2)*2+1;
if($page_3<=0){
        $page_3=0;
}
$query_3="select * from tb_mr_bccd limit $page_3,1";
$result3=$conn->prepare($query_3);                    //准备查询语句
$result3->execute();                                  //执行查询语句，并返回结果集
$res3=$result3->fetch(PDO::FETCH_ASSOC);
echo $res3['tb_bccd_name'];
?></a></td>
```

获取下一页超链接中 page 的值和下一页中的第一条数据，关键代码如下：

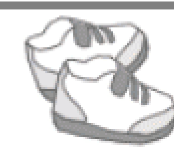
```
<td width="238" align="center" bgcolor="#FFFFFF">下一部：<a href="index.php?lmbs=明日编程词典&page=<?php
if($page=="" or $page<=1){
        echo 2;
}else if($page>=$page_count){
        echo $page_count;
}else{
        echo ceil(($page*2+1)/2);
}?>">
<?php
$page_4=($page)*2;
if($page_4<=0){
        $page_4=2;
}
if($page_4==$page_count*2){
        $page_4=$page_count*2-1;
}
$query_4="select * from tb_mr_bccd limit $page_4,1";
$result4=$conn->prepare($query_4);                    //准备查询语句
$result4->execute();                                  //执行查询语句，并返回结果集
$res4=$result4->fetch(PDO::FETCH_ASSOC);
echo $res4['tb_bccd_name'];
?></a></td>
```

PHP 上下分页中的其他内容已经在 2.6.2 节中介绍过了，这里不再赘述，完整程序代码参见配套资源中的相应内容。

2.8　技术提炼

2.8.1　自定义函数替换超长文本中特殊字符技术

在输出超长文本中的数据时，应用自定义函数 unhtml()实现对超长文本中的特殊字符进行替换。unhtml()函数的语法格式如下：




```
function unhtml($content){                                         //定义自定义函数的名称
    $content=htmlspecialchars($content);                           //转换文本中的特殊字符
    $content=str_replace(chr(13),"<br>",$content);                 //替换文本中的换行符
    $content=str_replace(chr(32)," ",$content);               //替换文本中的 
    $content=str_replace("[_[","<",$content];                      //替换文本中的大于号
    $content=str_replace(")_)",">",$content);                      //替换文本中的小于号
    $content=str_replace("|_|"," ",$content);                      //替换文本中的空格
    return trim($content);                                         //删除文本中首尾的空格
}
```

首先，通过 file_get_contents()函数读取超长文本中的数据，返回一个字符串变量$counter。unhtml()函数就是对这个字符串进行操作，替换字符串中的特殊字符，然后再执行其他操作。其关键代码如下：

```
<?php
include("function.php");                                           //调用指定的文件
if($page){
    $counter=file_get_contents("file/file.txt");                   //读取指定文件的内容
        $length=strlen(unhtml($counter));                          //获取字符串长度
        $page_count=ceil($length/1050);                            //统计每页的字符数
        $c=msubstr($counter,0,($page-1)*1050);                     //计算上页
        $c1=msubstr($counter,0,$page*1050);                        //计算当前页
    echo substr($c1,strlen($c),strlen($c1)-strlen($c));            //输出当前页的内容
  }
?>
```

file_get_contents()函数主要是用于将指定文件中的内容读入到一个字符串。如果操作系统支持内存映射技术，还可应用内存映射技术来增强其读入文件的性能。其语法格式如下：

```
string file_get_contents(string filename [, int use_include_path [, resource context]])
```

其中，参数说明如表 2.1 所示。

表 2.1 file_get_contents()函数的参数说明

参 数	说 明
filename	必要参数。用于指定某文件的路径及名称
use_include_path	可选参数。如果将该参数设置为True，PHP会尝试按照include_path标准
context	可选参数。设置提高文件流性能的一些选项

2.8.2 自定义函数控制超长文本的输出技术

在 PHP 超长文本分页显示中，控制超长文本中数据输出应用的是自定义函数 msubstr()。该函数的语法格式如下：

```
function msubstr($str,$start,$len){          //$str 指的是字符串，$start 指的是字符串的起始位
置，$len 指的是长度
    $tmpstr="";
    $strlen=$start+$len;                      //用$strlen 存储字符串的总长度（从字符串的起始
位置到字符串末尾的总长度）
```



```
    for($i=0;$i<$strlen;$i++){                    //通过 for 循环语句，循环读取字符串
        if(ord(substr($str,$i,1))>0xa0){          //如果字符串中首个字节的 ASCII 序数值大于
0xa0，则表示为汉字
            $tmpstr.=substr($str,$i,2);           //每次取出两位字符赋给变量$tmpstr，即等于
一个汉字
            $i++;                                 //变量自加 1
        }else{                      //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
            $tmpstr.=substr($str,$i,1);}
        }
    return $tmpstr;                               //输出字符串
}
```

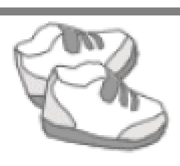
首先，应用 include 调用指定的包含文件 function.php；然后，应用 file_get_contents()函数读取指定文本文件中的数据，并应用自定义函数 unhtml()替换字符串中的特殊字符；最后，应用 msubstr()函数控制数据的输出。其关键代码如下：

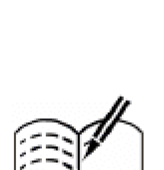
```
<?php
include("function.php");
if($page){
    $counter=file_get_contents("file/file.txt");//读取文本文件中的数据，返回一个字符串
    $length=strlen(unhtml($counter));            //计算字符串的长度，替换特殊字符
    $page_count=ceil($length/1050);              //对字符串进行分页处理
    $c=msubstr($counter,0,($page-1)*1050);       //通过自定义函数获取上一页中的数据
    $c1=msubstr($counter,0,$page*1050);          //通过自定义函数获取当前页中的数据
    echo substr($c1,strlen($c),strlen($c1)-strlen($c));   //获取到当前页中要输出的数据
  }
?>
```

2.8.3　连接数据库类的创建

要实现数据库中数据的分页显示，首先必须与数据库建立连接。这里通过一个自定义的类来实现数据库的连接，其中操作数据库使用 PDO 技术。该类存储于 02\01\system\system.class.inc.php 文件中，其代码如下：

```
class ConnDB{                                                   //数据库连接类
    var $dbtype;
    var $host;
    var $user;
    var $pwd;
    var $dbname;
    function ConnDB($dbtype,$host,$user,$pwd,$dbname){          //构造方法
        $this->dbtype=$dbtype;
    $this->host=$host;
        $this->user=$user;
        $this->pwd=$pwd;
        $this->dbname=$dbname;
    }
    function GetConnId(){                              //实现数据库的连接并返回连接对象
    if($this->dbtype=="mysql" || $this->dbtype=="mssql"){        //判断数据库类型
```





```
            $dsn="$this->dbtype:host=$this->host;dbname=$this->dbname";//定义连接语句
        }else{
            $dsn="$this->dbtype:dbname=$this->dbname";
        }
        try {
    $conn = new PDO($dsn, $this->user, $this->pwd);    //初始化一个 PDO 对象，创建数据库连接对象$pdo
            $conn->query("set names utf8");         //设置编码格式
        return $conn;                                //返回连接对象
        } catch (PDOException $e) {
        die ("Error!: " . $e->getMessage() . "<br/>");
        }
    }
}
```

2.8.4 操作数据库类的创建

数据库连接成功后，接下来进行数据库的操作。这里同样对 PDO 操作数据库的方法进行了封装。在 AdminDB 类中，可以执行数据的添加、更新、删除和查询操作。类文件存储于 2\01\system\system.class.inc.php 中，其代码如下：

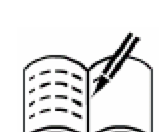
```
class AdminDB{                                              //数据库管理类
    function ExecSQL($sqlstr,$conn){
        $sqltype=strtolower(substr(trim($sqlstr),0,6));
        $rs=$conn->prepare($sqlstr);                  //准备查询语句
        $rs->execute();                               //执行查询语句，并返回结果集
        if($sqltype=="select"){
            $array=$rs->fetchAll(PDO::FETCH_ASSOC);//获取结果集中的所有数据
            if(count($array)==0 || $rs==false)
                return false;
            else
                return $array;
        }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
            if($rs)
                return true;
            else
                return false;
        }
    }
}
```

2.8.5 分页类的创建

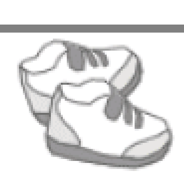
为了让分页技术的应用更加简单，可以将分页的方法写入一个类中，这样就可以在需要使用分页的地方直接调用这个类实现分页的功能。分页类同样存储于 2\01\system\system.class.inc.php 文件中，其代码如下：

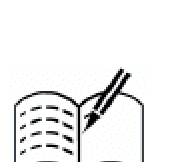
```
class SepPage{
```






```
    var $rs;
    var $pagesize;
    var $nowpage;
    var $array;
    var $conn;
    var $sqlstr;
    function ShowData($sqlstr,$conn,$pagesize,$nowpage){          //定义方法
        if(!isset($nowpage) || $nowpage=="")                       //判断变量值是否为空
            $this->nowpage=1;                                      //定义起始页
        else
            $this->nowpage=$nowpage;
        $this->pagesize=$pagesize;                         //定义每页输出的记录数
        $this->conn=$conn;                                 //连接数据库返回的标识
        $this->sqlstr=$sqlstr;                             //执行的查询语句
        $offset=($this->nowpage-1)*$this->pagesize;
        $sql=$this->sqlstr." limit $offset, $this->pagesize";
        $result=$this->conn->prepare($sql);                //准备查询语句
        $result->execute();                        //执行查询语句，并返回结果集
        $this->array=$result->fetchAll(PDO::FETCH_ASSOC);  //获取结果集中的所有数据
        if(count($this->array)==0 || $this->array==false)
            return false;
        else
            return $this->array;
    }
    function ShowPage($contentname,$utits,$anothersearchstr,$anothersearchstrs,$class){
        $str="";
        $res=$this->conn->prepare($this->sqlstr);      //准备查询语句
        $res->execute();                               //执行查询语句，并返回结果集
        $this->array=$res->fetchAll(PDO::FETCH_ASSOC);//获取结果集中的所有数据
        $record=count($this->array);                   //统计记录总数
        $pagecount=ceil($record/$this->pagesize);      //计算共有几页
        $str.=$contentname." ".$record." ".$utits." 每页 
  ".$this->pagesize." ".$utits."  第  ".$this->nowpage."  页 / 共
 ".$pagecount." 页";
        $str.="    ";
        if($this->nowpage!=1)
            $str.="<a href=".$_SERVER['PHP_SELF']."
  ?page=1&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs."
class=".$class.">首页</a>";
        else
            $str.="<font color='#555555'>首页</font>";
        $str.=" ";
        if($this->nowpage!=1)
            $str.="<a href=".$_SERVER['PHP_SELF']."?
  page=".($this->nowpage-1)."
  &page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">上一页</a>";
        else
            $str.="<font color='#555555'>上一页</font>";
        $str.=" ";
```





Note

```
			if($this->nowpage!=$pagecount)
				$str.="<a href=".$_SERVER['PHP_SELF']."?page=".($this->nowpage+1)."&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">下一页</a>";
			else
				$str.="<font color='#555555'>下一页</font>";
			$str.=" ";
			if($this->nowpage!=$pagecount)
				$str.="<a href=".$_SERVER['PHP_SELF']."?page=".$pagecount."&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">尾页</a>";
			else
				$str.="<font color='#555555'>尾页</font>";
			if(count($this->array)==0 || $this->array==false)
				return "无数据！ ";
			else
				return $str;
		}
	}
```

在这个分页类中，包含两个方法，即 ShowData($sqlstr,$conn,$pagesize,$nowpage)和 ShowPage($contentname,$utits,$anothersearchstr,$anothersearchstrs,$class)。其中，ShowData()方法根据参数传递的 SQL 语句，完成数据的分页查询操作，并且返回查询结果；在 ShowPage()方法中，根据参数传递的值定义分页超链接的输出内容。

2.8.6 类的实例化

类创建成功之后，要想在程序中调用类中的方法完成指定的操作，必须先完成类的实例化操作，通过类实例化返回的对象才可以调用指定的方法。有关类的实例化操作在本模块中都统一存储在 2\01\system\system.inc.php 文件中，其代码如下：

```
<?php
require("system.smarty.inc.php");			//包含 Smarty 配置类
require("system.class.inc.php");			//包含数据库连接和操作类
$connobj=new ConnDB("mysql","localhost","root","111","db_pagination");		//数据库连接类实例化
$conn=$connobj->GetConnId();			//执行连接操作，返回连接标识
$admindb=new AdminDB();				//数据库操作类实例化
$seppage=new SepPage();				//分页类实例化
$usefun=new UseFun();				//使用常用函数类实例化
$smarty=new SmartyProject();			//调用 smarty 模板
function unhtml($params){
   extract($params);
   $text=$content;
   global $usefun;
   return $usefun->UnHtml($text);
}
$smarty->register_function("unhtml","unhtml");//注册模板函数
?>
```

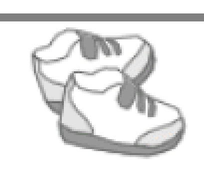
第 3 章

上传与下载模块

（自学视频、源程序：配套资源\mr\3\）

文件的上传与下载是一个完整的网站不可或缺的功能之一，如果不能提供这样的功能，浏览者便会觉得这个网站的开发不是很完美。本章将全面介绍PHP的上传、下载技术，包括从 php.ini 文件的配置，到上传函数的使用，到最后的动手实践。通过本章的学习，读者能够学到以下内容。

- 文件目录及文件的相关操作
- 文件的上传
- 文件的下载
- 文件重名自动检测
- 利用下载码实现文件搜索
- 如何限制用户的空间使用大小
- 文件上传的相关配置





Note

3.1 上传、下载模块概述

3.1.1 模块概述

使用过 ASP 的读者一定会对其上传功能记忆犹新。要在 ASP 中实现上传，需要使用 Active 组件完成，或者编写一大段代码实现无组件上传。这对于新手来说，难免令人望而生畏，即便具有一定的基础也会有一定的难度。PHP 在这方面要比 ASP 方便得多，只要一个上传函数就能够完全实现。如果是针对大文件上传或下载，也只需在 php.ini 配置文件中设置几个参数值即可。在本模块中将对 PHP 的上传和下载功能的实现进行系统、全面的讲解。

3.1.2 功能结构

为了便于读者更好地学习，下面给出上传、下载模块的功能结构流程图。

在上传、下载模块中，用户需要先注册为本模块用户，然后才能登录模块进行相关操作。用户未登录时，身份为游客，可通过下载码下载未公开文件。用户成功登录后，系统将自动判别访客的级别，级别不同，分配的权限也不同。根据级别的不同，可以执行上传新文件、删除已上传个人文件等操作，还可以更改文件公布状态等。

上传、下载模块的功能结构如图 3.1 所示。

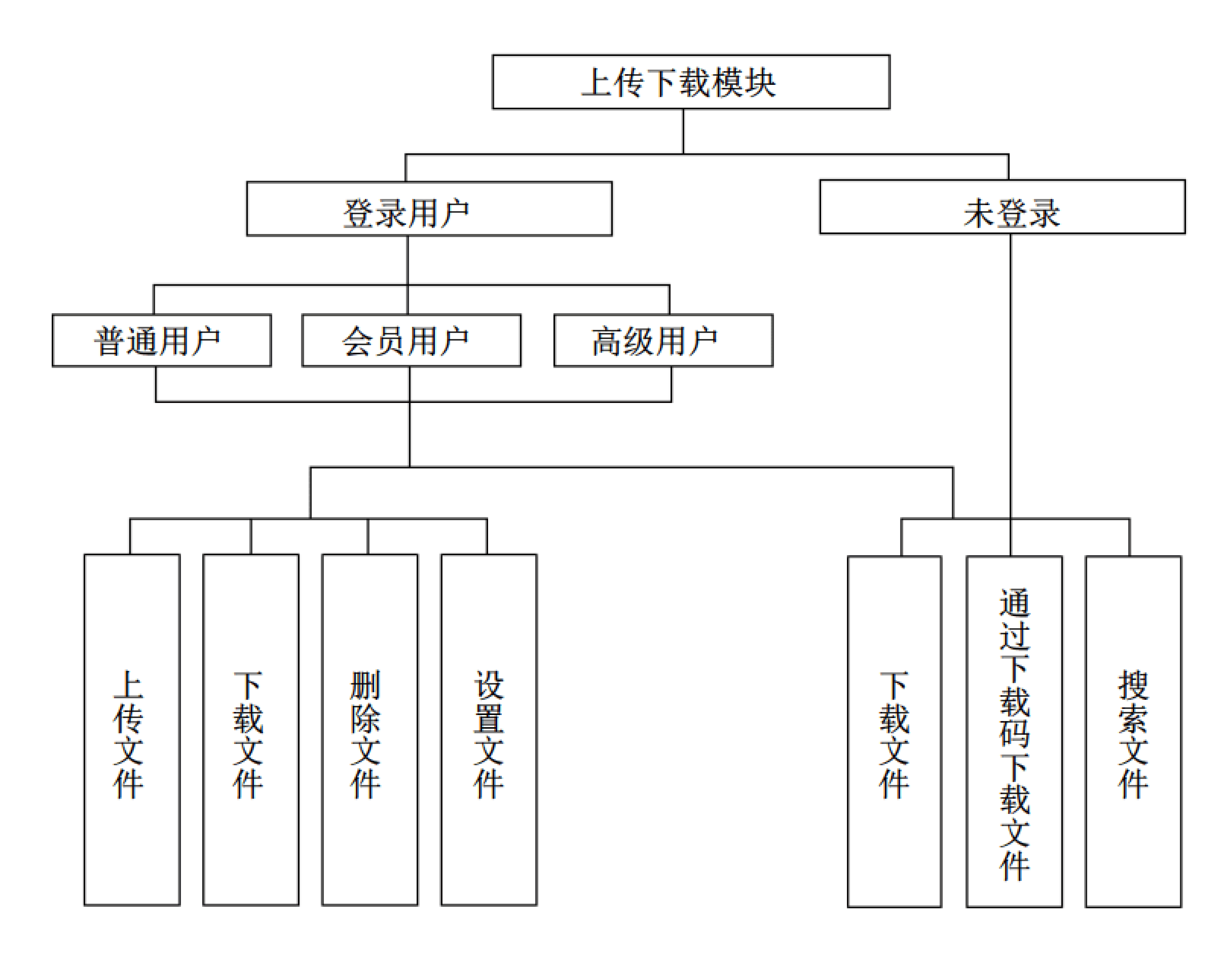


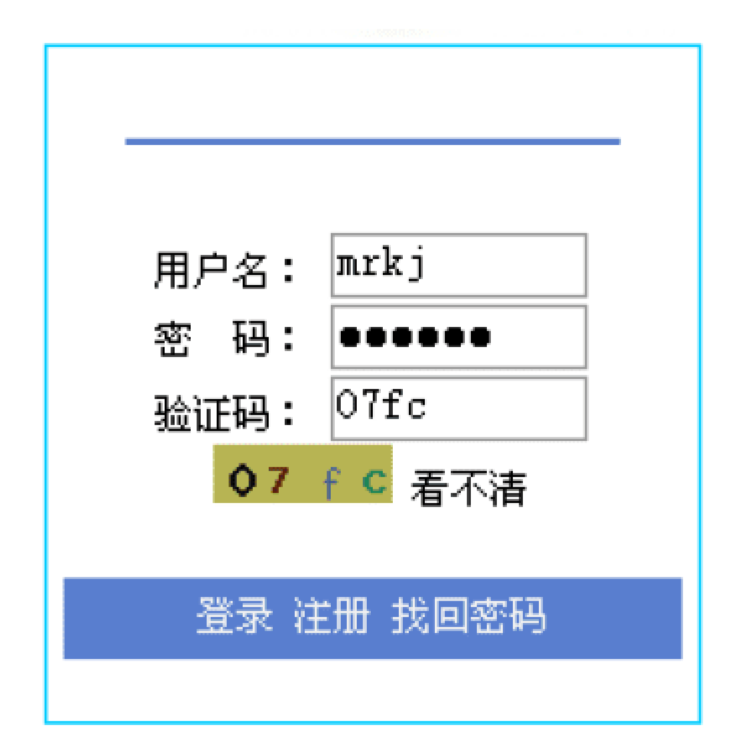
图 3.1　上传下载模块的功能结构图

3.1.3 程序预览

上传、下载模块由多个页面组成，下面列出几个典型的页面，其他的参见配书光盘中的程序。



上传、下载模块的登录页面如图 3.2 所示。如果想下载自己上传的文件，则首先执行登录操作，登录成功后才可以在个人信息显示处单击“下载文件”超链接进入下载页。



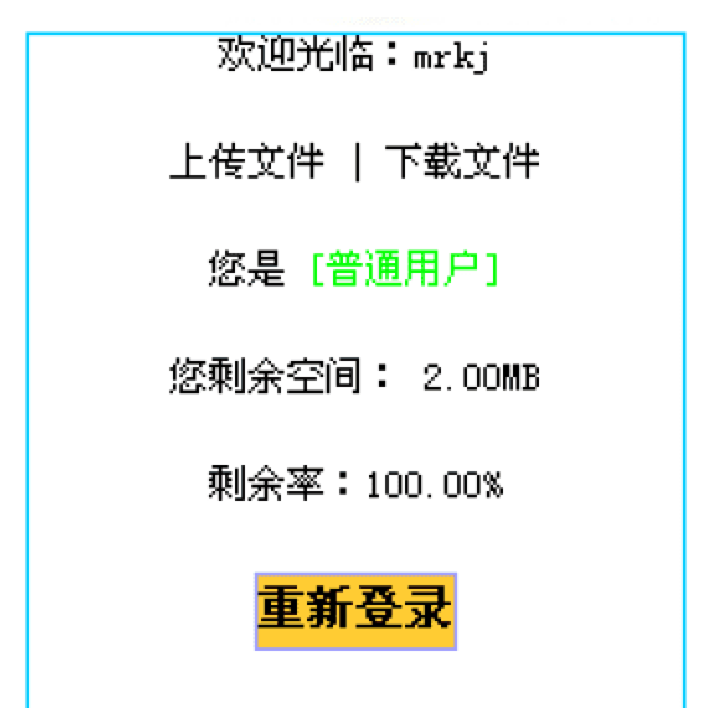


图 3.2　用户登录页面

上传、下载模块的主页如图 3.3 所示。如果用户身份为游客，则可通过下载码下载未公开文件；如果成功登录，则可以执行上传新文件、删除已上传个人文件等操作，还可以更改文件公布状态等。



图 3.3　上传、下载模块的主页

上传文件页面如图 3.4 所示，主要是用于上传新文件到系统分配的空间内。

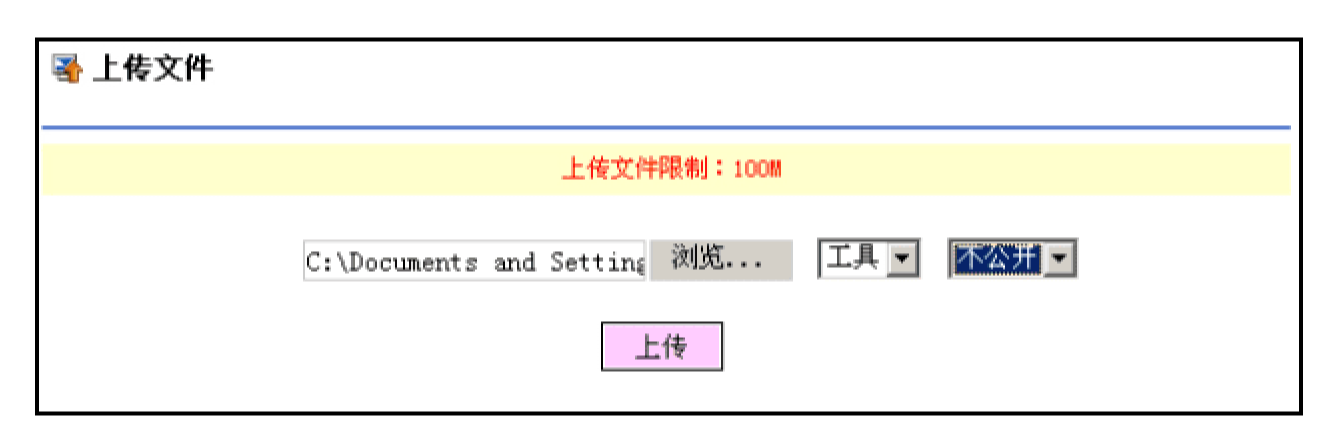
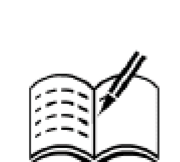


图 3.4　上传文件页面

下载文件页面如图 3.5 所示。顾名思义，该页面主要是展示已上传文件的相关信息，用户可以通过单击该页面上文件名称的超链接下载所需文件。



Note





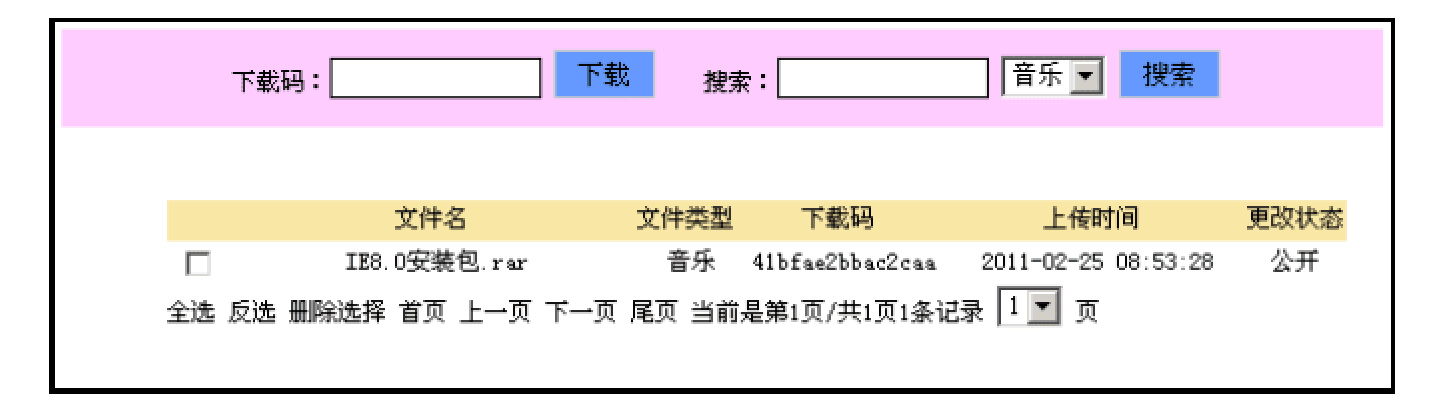


图 3.5　下载文件页面

当前用户为游客身份时，也可以通过文件的下载码下载对应的上传文件，如图 3.6 所示。

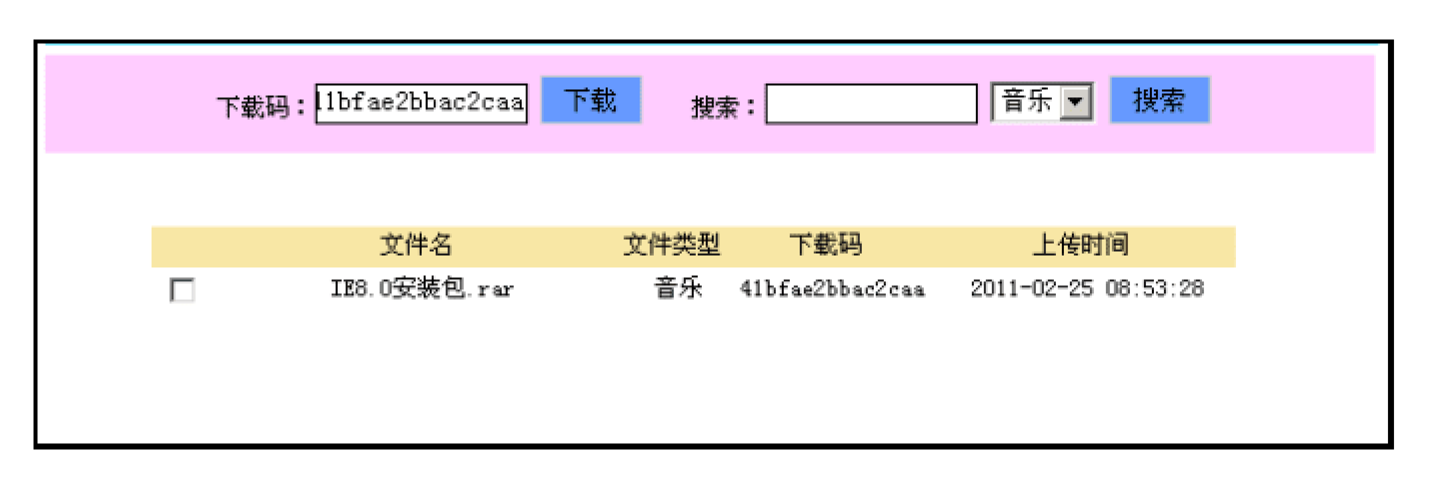


图 3.6　通过下载码下载文件

3.2　数据库设计

下面通过实现一个上传、下载模块，学习上传、下载技术在实际中的应用。首先来看数据库的设计和连接。

3.2.1　数据库分析

首先定义上传下载模块，使用 MySQL 数据库，名称是 db_netu。其中包含 3 个表，分别是 tb_member（用户信息表）、tb_upfile（上传文件表），以及 tb_uptype（文件类型表），如图 3.7 所示。其中，tb_member 存储用户的相关信息；tb_upfile 是整个上传、下载模块的核心数据表，记录了上传、下载文件的相关信息；tb_uptype 主要记录上传文件的种类。


服务器: localhost ▸ 数据库: db_netu

结构 SQL 搜索 查询 导出 Import 操作

表	类型	整理	说明
tb_member	MyISAM	gb2312_chinese_ci	用户列表
tb_upfile	MyISAM	gb2312_chinese_ci	上传文件列表
tb_uptype	MyISAM	gb2312_chinese_ci	文件类型列表

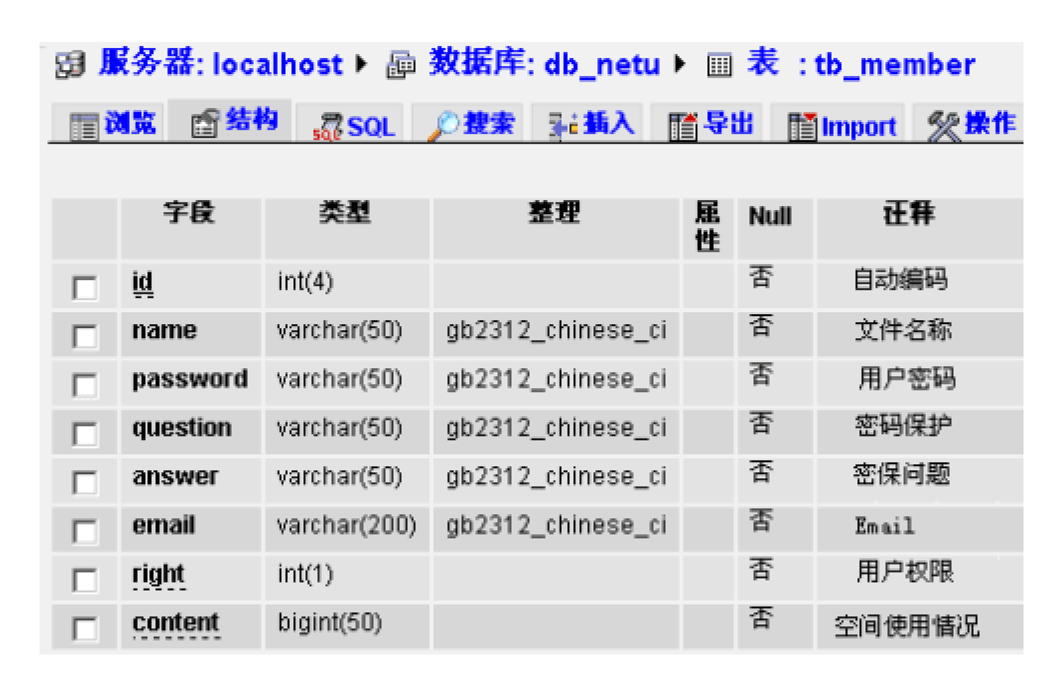
图 3.7　db_netu 数据库

3.2.2　创建数据表

1. tb_member 表

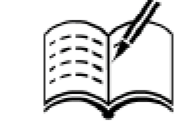
tb_member 表记录了注册用户，包括用户名、密码、密保问题、密保答案、用户等级权限以及用户个人空间使用情况等信息。tb_member 表的结构及说明如图 3.8 所示。





	字段	类型	整理	属性	Null	注释
	id	int(4)			否	自动编码
	name	varchar(50)	gb2312_chinese_ci		否	文件名称
	password	varchar(50)	gb2312_chinese_ci		否	用户密码
	question	varchar(50)	gb2312_chinese_ci		否	密码保护
	answer	varchar(50)	gb2312_chinese_ci		否	密保问题
	email	varchar(200)	gb2312_chinese_ci		否	Email
	right	int(1)			否	用户权限
	content	bigint(50)			否	空间使用情况

图 3.8　tb_member 表结构及说明



2. tb_upfile 表

tb_upfile 表记录了用户上传的文件信息，包括文件名、文件类型、文件下载码等。tb_upfile 表的结构及说明如图 3.9 所示。



字段	类型	整理	说明
id	int(4)		自动编号
filename	varchar(200)	gb2312_chinese_ci	文件名称
filepath	varchar(200)	gb2312_chinese_ci	保存路径
filetype	varchar(100)	gb2312_chinese_ci	文件类型
upauthor	varchar(200)	gb2312_chinese_ci	上传用户
uptime	timestamp		上传时间
chkdownnum	varchar(20)	gb2312_chinese_ci	下载码
ispub	int(1)		是否公开
filesize	bigint(20)		文件大小

图 3.9　tb_upfile 表结构及说明

3. tb_uptype 表

tb_uptype 表规定了上传文件的类型，用户可以将文件上传到不同的类型中，以方便日后查找。tb_uptype 表的结构及说明如图 3.10 所示。



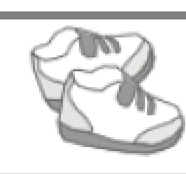
字段	类型	整理	额外	说明
id	int(4)		auto_increment	自动编号
genrename	varchar(200)	gb2312_chinese_ci		类型名称

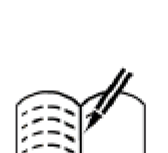
图 3.10　tb_uptype 表结构及说明

3.2.3　数据库连接文件

PHP 连接数据库并非难事，但是在每次应用数据库连接时都重复输入类似代码，无疑会浪费大量的时间，同时也降低了代码的书写效率和可读性。所以，本模块中将数据库的连接方法封装到一个类中，这样在应用数据库连接操作时，调用该类文件即可。conn\conn.php 文件的关键代码如下：

```
<?php
```





Note

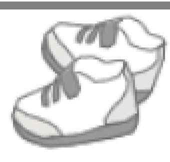
```
class opmysql{
	private $host = 'localhost';		/服务器地址
	private $name = 'root';			//登录账号
	private $pwd = '111';			//登录密码
	private $dBase = 'db_netu';		//数据库名称
	private $conn = '';			//数据库链接资源
	private $result = '';			//结果集
	private $msg = '';			//返回结果
	private $fields;			//返回字段
	private $fieldsNum = 0;			//返回字段数
	private $rowsNum = 0;			//返回结果数
	private $rowsRst = '';			//返回单条记录的字段数组
	private $filesArray = array();		//返回字段数组
	private $rowsArray = array();		//返回结果数组
	//初始化类
	function __construct($host='',$name='',$pwd='',$dBase=''){
		if($host != '')
			$this->host = $host;
		if($name != '')
			$this->name = $name;
		if($pwd != '')
			$this->pwd = $pwd;
		if($dBase != '')
			$this->dBase = $dBase;
		$this->init_conn();
	}
	//链接数据库
	function init_conn(){
		$this->conn=@mysql_connect($this->host,$this->name,$this->pwd);
		@mysql_select_db($this->dBase,$this->conn);
		mysql_query("set names gb2312");
	}	private $dBase = 'db_netu';		//数据库名称
						//省略部分代码
```

另外，该自定义类中还包含了一些常用的数据库操作方法。具体方法和相关说明参见配书光盘中的源程序，这里不再赘述。

3.3 用户注册

3.3.1 注册模块概述

本上传、下载模块是一个公开面向用户的个人网络存储空间程序， 自助式的用户注册是其功能模块之一。新用户可以通过单击登录模块中的“注册”超链接来进入自助式用户注册页面，如图 3.11 所示。





电子相册注册页面

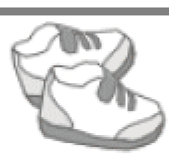
注册名称：	mrkj	恭喜您，该用户名可以使用！
注册密码：	••••••	密码符合要求。密码强度：弱
确认密码：	••••••	密码输入正确
密保问题：	111	
密保答案：	111	
Email：	mingrosft@mingrisoft.com	
注册	详细信息	

图 3.11　用户注册页面

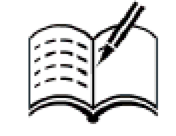
3.3.2　注册页面设计

注册页面的设计没有太多实质性的理论可言，其主要是通过表单来获取用户注册的相关信息，将数据提交到 register_chk.php 文件中，完成用户注册信息的处理。注册页面文件 register.php 的相关代码如下：

```
<?php
        session_start();
        header('Content-Type:text/html;charset=gb2312');
?>
<table id="regfm" width="500" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr>
    <td height="25" colspan="3" align="center" valign="middle">电子相册注册页面</td>
    </tr>
  <tr>
    <td width="100" height="25" align="right" valign="middle">注册名称：</td>
    <td width="200" height="25" align="left" valign="middle"> 
<input id="regname" name="regname" type="text" class="txt" /></td>
    <td height="25"><div id="namediv" class="regdiv">名称由英文字母和数字及下划线组成</div></td>
    </tr>
  <tr>
    <td width="100" height="25" align="right" valign="middle">注册密码：</td>
    <td width="200" height="25" align="left" valign="middle"> 
<input id="regpwd1" name="regpwd1" type="password" class="txt" /></td>
    <td height="25"><div id="pwddiv1" class="regdiv">请输入密码</div></td>
    </tr>
  <tr>
    <td width="100" height="25" align="right" valign="middle">确认密码：</td>
    <td width="200" height="25" align="left" valign="middle"> 
<input id="regpwd2" name="regpwd2" type="password" class="txt" /></td>
    <td height="25"><div id="pwddiv2" class="regdiv">确认密码</div></td>
    </tr>
</table>
</table>
<div id="morediv" style="display:non;">
<!--省略部分代码-->
</table>
```




```
</div><center>
<div style=" text-align:center;text-align:center; width:50%; height:25px; line-height:25px;">
<button id="regbtn" class="btn" disabled="disabled">注册</button> 
<button id="morebtn" class="btn">详细信息</button></div></center>
```


Note

3.3.3 Ajax 无刷新技术验证注册信息

新用户注册账号时，如果在每次提交注册信息之后才返回不符合要求的错误信息，不但会降低执行效率（用户之前填写的所有注册信息都会随着操作失败而消失），也会消磨用户的耐心。为了让用户及时得到反馈信息，注册模块采用目前比较流行的 Ajax 技术，对注册信息执行无刷新验证操作。这里以验证注册用户名为例进行介绍。该过程是通过 js 文件类目录下的 register.js 文件来实现的。其关键代码如下：

```
function $(id){
    return document.getElementById(id);                //获取标签 ID
}
$('regname').onblur = function(){
        name = $('regname').value;
        if(cname1 == 'yes'){
            xmlhttp.open('get','chkname.php?name='+name,true);
            xmlhttp.onreadystatechange = function(){
                if(xmlhttp.readyState == 4){
                    if(xmlhttp.status == 200){
                        var msg = xmlhttp.responseText;
                        if(msg == '1'){                //判断用户名是否被占用
        $('namediv').innerHTML="<font color=green>恭喜您，该用户名可以使用!</font>";
                            cname2 = 'yes';
                        }else if(msg == '2'){
                            $('namediv').innerHTML="<font color=red>用户名被占用！ </font>";
                            cname2 = '';
                        }else{
                            $('namediv').innerHTML="<font color=red>"+msg+"</font>";
                            cname2 = '';
                        }
                    }
                }
            }
            xmlhttp.send(null);
            chkreg();
        }
    }
```

在 register.js 文件中，无刷新调用 chkname.php 文件，对注册用户名进行验证。根据 chkname.php 文件的返回值进行判断，该用户名是否可用。chkname.php 文件的关键代码如下：

```
<?php
```






```
session_start();                                    //开启会话
include_once "conn/conn.php";                       //加载数据库配置文件
$reback = '0';                                      //定义验证返回值
$sql = "select * from tb_member where name='".$_GET['name']."'";
$num = $conne->getRowsNum($sql);                    //执行数据库查询操作
if($num == 1){                                      //判断是否存在匹配结果
        $reback = '2';                              //如果存在则将验证返回值赋值 2
}else if($num == 0){
        $reback = '1';                              //如果不存在则将验证返回值赋值 1
}else{
        $reback = $conne->msg_error();              //输出错误信息
}
echo $reback;                                       //输出验证返回值
?>
```

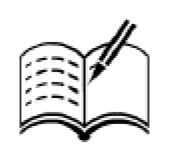
3.3.4　注册信息处理

创建 register_chk.php 文件，获取注册页面 register.php 中提交的注册信息，最终将用户的注册信息添加到指定的数据表中，同时为新注册用户创建一个属于自己的存储文件目录。注册信息处理文件 register_chk.php 的相关代码如下：

```
<?php
        session_start();
        header('Content-Type:text/html;charset=gb2312');
        include_once 'conn/conn.php';
        $reback = '0';
        $sql = "insert into tb_member(name,password,question,answer,email)
values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."',
'".$_GET['answer']."','".$_GET['email']."')";
        $num = $conne->uidRst($sql);
        if($num == 1){
                $_SESSION['name'] = $_GET['name'];
                $username=$_SESSION['name'];
                $dirname="../upfile/$username";             //创建一个和用户名相同的文件夹
                mkdir($dirname,0700);
                $reback = '1';
                $file = 'index.txt';
                $newfile="../upfile/$username/index.php";
                if (!copy($file, $newfile)) {
            echo "failed to copy $file...\n";
        }
        }else{
                $reback = $conne->msg_error();
        }
        echo $reback;
?>
```



3.4 用 户 登 录



Note

3.4.1 登录模块概述

上传、下载模块通过用户登录实现个人空间的划分、控制上传文件的公开或隐藏，以及对个人上传文件的管理。登录模块的运行效果如图 3.12 所示。



图 3.12 登录模块运行效果

登录模块中包含用户登录、新用户注册、找回密码等功能；同时输出了用户的权限、剩余空间大小、剩余空间百分比以及“上传文件”和“下载文件”超链接。由于篇幅所限，下面只对用户登录模块中的一些特殊功能进行详细讲解，其完整内容参见配书光盘源代码。

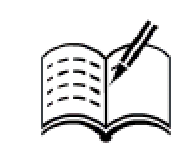
3.4.2 用户权限输出以及空间使用计算

用户登录后，系统将输出用户的权限以及用户个人空间的使用情况。下面介绍如何判断登录用户的权限，以及计算空间的使用情况。

（1）在 index.php 中，根据 SESSION 变量传递的登录用户名，从数据库中读取该用户的个人信息，对 tb_member 数据表中 right 字段的值进行判断，进而输出用户的权限。其关键代码如下：

```
<?php
$rightsql = "select * from tb_member where name = '".$_SESSION['name']."'";   //查询语句
$rightarr = $conne->getRowsRst($rightsql);            //返回查询结果集，判断用户权限
$conne->close_rst();
if($rightarr['right']==2) {                           //如果返回查询值为 2，判断为高级用户
    echo "<span class=\"vip1\">[高级用户]</span>";
}else if($rightarr['right']==1) {
    echo "<span class=\"vip2\">[会员用户]</span>"; //如果返回查询值为 1，判断为会员用户
}else{
        echo "<span class=\"vip3\">[普通用户]</span>";          //否则判断为普通会员
}
?>
```





Note

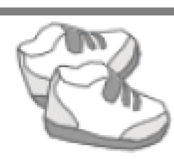
（2）计算用户的空间使用情况。首先，从数据库中读取权限字段中的数值，判断当前用户的登录权限，通过用户的权限来设定其使用空间大小。然后，从数据库中读取该用户的空间使用情况，将其分配空间的大小与该用户已使用空间大小进行差值运算。最后，将取得的结果进行格式化输出，显示当前登录用户的剩余空间和剩余空间的百分比。相关代码如下：

```
<p>您剩余空间：
<?php
if($rightarr['right']==2) {                    //如果权限为高级用户，则为其分配空间 5MB
    if(5000000-$rightarr['content']<=1000) {       //判断用户空间是否不足
        echo "已满！";
    }else {
        $formatted=(5000000-$rightarr['content'])/1000000; //对剩余空间大小进行换算
        $forstr=sprintf("%-01.2f",$formatted);   //对计算的空间大小数值进行格式化
        echo $forstr."MB";
        echo "<p>剩余率:".sprintf("%-01.2f",((1-$rightarr['content']/5000000)*100))."%</p>";
    }                                            //对用户的剩余空间以百分比的格式输出
}else if($rightarr['right']==1) {
    if(3000000-$rightarr['content']<=1000) {
        echo "已满！";
      }else{
          $formatted=(3000000-$rightarr['content'])/1000000;
          $forstr=sprintf("%-01.2f",$formatted);
          echo $forstr."MB";
          echo "<p>剩余率:".sprintf("%-01.2f",((1-$rightarr['content']/3000000)*100))."%</p>";
              }
    }else {
          if(2000000-$rightarr['content']<=1000) {
            echo "已满！";
          }else{
    $formatted=(2000000-$rightarr['content'])/1000000;
            $forstr=sprintf("%-01.2f",$formatted);
            echo $forstr."MB";
            echo "<p> 剩 余 率： ".sprintf("%-01.2f",((1-$rightarr['content']/2000000)*100))."%</p>";
                }
          }
?>
```

3.4.3　为用户划分个人文件目录

为注册的每一个用户创建一个与其注册名同名的个人文件目录，这样不但可以方便管理员对个人文件的管理，同时也避免了所有用户上传文件到同一文件目录下的弊端。

创建个人文件目录的操作在用户的注册处理页 register_chk.php 中完成。首先，获取表单提交的用户注册信息；然后，执行添加语句，将注册信息添加到指定的数据表；最后，通过 mkdir()函数在服务器指定的目录下创建一个与用户名相同的文件夹。其关键代码如下：





```
<?php
        session_start();
        header('Content-Type:text/html;charset=gb2312');
        include_once 'conn/conn.php';
        $reback = '0';
        $sql = "insert into tb_member(name,password,question,answer,email)
values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."',
'".$_GET['answer']."','".$_GET['email']."')";
        $num = $conne->uidRst($sql);
        if($num == 1){
            $_SESSION['name'] = $_GET['name'];
                $username=$_SESSION['name'];
                $dirname="../upfile/$username";        //设置文件夹的路径及名称
                mkdir($dirname,0700);                  //创建一个和用户名相同的文件夹
$reback = '1';
```

3.4.4　防止用户文件目录的非法访问

为了避免某些用户通过不正当方式获取其他用户文件夹的地址，从而直接通过地址访问其他用户文件夹，获得相关的内容并下载其他用户文件。可在为每个用户创建文件夹的同时，在个人文件夹下创建一个 index.php 文件，以此来屏蔽直接通过地址栏访问文件夹下的内容。

首先，在 login 文件夹下创建一个 index.txt 文本文件。代码如下：

```
<?php
        echo "请使用正确的访问路径！";
?>
<p>
点击<a href="../../index.php">这里</a>返回                  //单击返回首页
</p>
```

然后，在注册信息处理页 register_chk.php 中，完成用户注册信息的添加之后，将 index.txt 文本文件复制到新创建的用户个人文件夹下，并且命名为 index.php。其关键代码如下：

```
$file = 'index.txt';
$newfile="../upfile/$username/index.php";                  //设置复制新路径名称
if (!copy($file, $newfile)) {                              //将 index.txt 复制到新路径
    echo "failed to copy $file...\n";
}
}else{
            $reback = $conne->msg_error();
        }
        echo $reback;
?>
```

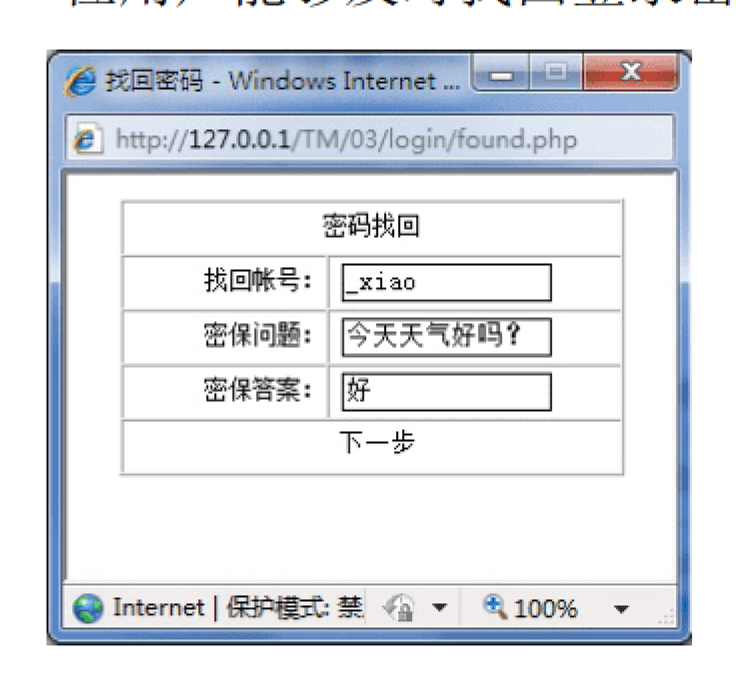


Note

3.5　找 回 密 码

3.5.1　找回密码概述

长时间未登录的用户可能会忘记网站的登录密码；另外，一些用户的密码也可能会被他人非法更改。为此，必须创建一个密码找回模块，为个人用户取回密码提供一个有效平台，从而保证每一位用户能够及时找回登录密码。密码找回模块的运行效果如图 3.13 所示。



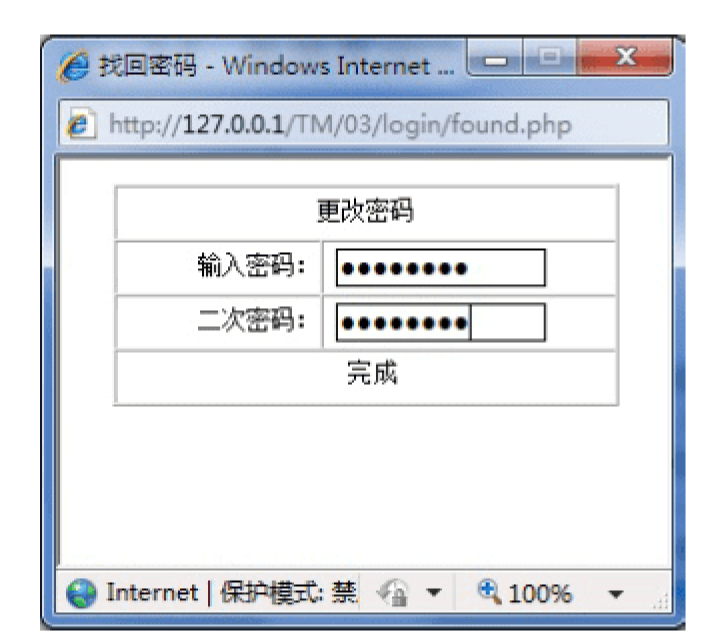


图 3.13　密码找回模块的运行效果

在本模块中，找回密码采用的是密保问题与密保答案比较的方式。首先用户输入注册的用户名；然后输入密保问题和密保答案；最后对用户提交的问题和答案与数据库中存储的数据进行比较，如果问题和答案都正确，则该用户可以执行密码的修改操作，重新设置新的密码。

3.5.2　找回密码页面设计

在找回密码页面中，关键是创建表单，提交注册的用户名、密保问题和密保答案；同时还嵌入了两个不同 ID 值的 div 层，通过控制 div 层样式的隐藏和展开来实现密码找回表单和更改密码表单之间的切换操作。找回密码文件 found.php 的关键代码如下：

```
<!--找回密码部分-->
 <div id="foundinfo">
<table width="250" border="1" align="center" cellpadding="0" cellspacing="0">
<tr>
    <td height="25" colspan="2" align="center" valign="middle">密码找回</td>
  </tr>
  <tr>
    <td width="100" height="25" align="right" valign="middle">找回账号：</td>
    <td height="25"> 
<input id="foundname" type="text" style=" width: 100px; height:15px; border:1px #000000 
solid;" /></td>
  </tr>
 <!--省略部分代码-->
  <tr>
    <td height="25" colspan="2" align="center" valign="middle"> 
```





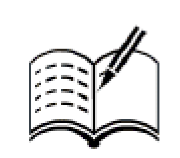
```
<button id="step1" class="btn">下一步</button></td>
  </tr>
</table>
</div>
<!--更改密码部分-->
<div id="changepwd" style="display:none;">
<table width="250" border="1" align="center" cellpadding="0" cellspacing="0">
<tr>
    <td height="25" colspan="2" align="center" valign="middle">更改密码</td>
  </tr>
  <tr>
    <td width="100" height="25" align="right" valign="middle">输入密码：</td>
    <td height="25"> 
<input id="pwd1" type="password" style=" width: 100px; height:15px; border:1px #000000 solid;" /></td>
  </tr>
  <tr>
     <td width="100" height="25" align="right" valign="middle">二次密码：</td>
    <td height="25"> 
<input id="pwd2" type="password" style=" width: 100px; height:15px; border:1px #000000 solid;" /></td>
  </tr>
  <tr>
    <td height="25" colspan="2" align="center" valign="middle"> 
<button id="step2" class="btn">完成</button></td>
  </tr>
</table>
</div>
```

在 found.php 文件中，载入了 js\found.js 脚本文件，通过 Ajax 无刷新技术调用 found_chk.php 文件，对用户提交的密保问题和密保答案进行验证，并且根据 found_chk.php 文件中的返回值控制 div 标签的隐藏和显示。found.js 脚本文件的代码如下：

```
    $('step1').onclick = function(){
        if($('foundname').value != '' && $('question').value != '' && $('answer').value != ''){
            xmlhttp.open('get','found_chk.php?act=sel&foundname='+$('foundname')
.value+'&question='+$('question').value+'&answer='+$('answer').value,true);        //验证找回密码预设信息
            xmlhttp.onreadystatechange = function(){
                if(xmlhttp.readystate == 4 && xmlhttp.status == 200){
                    msg = xmlhttp.responseText;
                    if(msg == '1'){
                        $('foundinfo').style.display = 'none';        //更改 display 属性隐藏 foundinfo 层
                        $('changepwd').style.display = '';            //更改 display 属性显示 changepwd 层
                    }else{
                        alert('填写信息错误！');
                    }
                }
            }
```



```
                    xmlhttp.send(null);
            }
    }
```



3.5.3　找回密码处理

创建找回密码处理文件 found_chk.php。该处理文件主要包括两个不同的处理方法，即验证密码保护问题和更改密码。首先通过 URL 获取 act 的参数值来判断用户执行的操作；然后根据选择的 act 参数来执行相应的方法，对用户提交的密保问题和密保答案进行验证，或者完成对用户提交新密码的更新操作。其关键代码如下：

```
<?php
    header('Content-Type:text/html;charset=gb2312');
    include_once 'conn/conn.php';
    $act = $_GET['act'];
    $reback = '0';
    if($act == 'sel'){                                          //判断用户选择的操作
        $name = $_GET['foundname'];                             //获取找回密码用户名
        $question = $_GET['question'];                          //找回密码问题
        $answer = $_GET['answer'];                              //找回密码
$sql="select * from tb_member where name='".$name."' and question='".$question."' and answer='".$answer."'";
        $num = $conne->getRowsNum($sql);
    }else if($act == 'up'){
        $name = $_GET['foundname'];
        $pwd = $_GET['pwd'];
        $sql = "update tb_member set password = '".md5($pwd)."' where name = '".$name."'";
        $num = $conne->uidRst($sql);                            //执行数据库更新操作
    }
    if($num != ''){
        $reback = '1';                                          //返回值为 1
    }else{
        $reback = $sql;
    }
    echo $reback;
?>
```

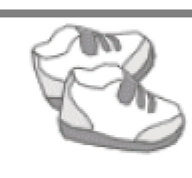
说明：

用户提交的新密码首先会经 MD5()函数加密，然后再被插入数据库中。

3.6　文件上传

3.6.1　文件上传概述

作为网络移动存储工具，上传是必不可少的功能之一。用户可以通过登录个人账户，单击“上传文件”超链接，进入文件上传页，将自己想要上传的工具、文档、音频或视频





Note

等多种类型的文件上传到个人空间中。文件上传页面的运行效果如图 3.14 所示。



图 3.14　文件上传页面的运行效果

3.6.2　文件上传表单设计

文件上传要通过表单中的文件域来提交，在处理页中通过$_FILES[]全局变量和 move_upload_file()函数完成文件的上传操作。

上传文件的表单存储于 upfile.php 文件中。首先，创建 form 标记，指定 POST()方法提交，设置 enctype 属性等于"multipart/form-data"，设置处理页 upfile_chk.php。然后，创建文件域，选择上传文件。接着，创建下拉列表框展示文件类型，将从数据库读取的文件类型作为下拉列表框的值。接下来，创建下拉列表框，设置文件权限。最后，创建提交按钮，通过 onClick()事件调用 JavaScript 脚本中的方法验证提交数据，完成提交操作。upfile.php 文件的关键代码如下：

```
…
<!--上传表单，处理页为 upfile_chk.php-->
<form method="post" enctype="multipart/form-data" action="?act=upfile">
<!--列表项-->
<ul id="change">
<!--上传表单。注意：name 值为一个数组-->
<li id="addordel">
<input id="upname[]" name="upname[]" type="file" style=" width: 200px; background-color:#f0f0f0; " /> 
<!--文件类别下拉列表，name 也是数组-->
<select id="foundtype[]" name="foundtype[]">
<?php
    $typesql = "select genrename from tb_uptype";        //查询语句
    $typearr = $conne->getRowsArray($typesql);            //返回查询结果集
    $conne->close_rst();                                  //清空数组
```

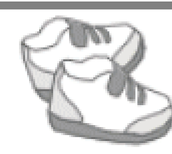
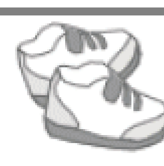
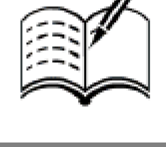



```
        foreach($typearr as $value){                    //循环输出类别列表
    ?>
            <option value="<?php echo $value['genrename']; ?>"><?php echo $value['genrename']; ?></option>
    <?php
        }
    ?>
    </select> 
    <!--判断文件是否公开。0，不公开；1，公开-->
    <select id="ispub[]" name="ispub[]">
        <option value="0" selected="selected">不公开</option>
        <option value="1">公开</option>
    </select> 
    <!--删除上传域和添加上传域-->
    </li>
    </ul>
    <input id="upbtn" name="upbtn" type="submit" value="上传"
    onClick="return chkfile()" style=" background-color:#f0f0f0; border: 1px #CCCCCC solid;" />
    </form>
```

3.6.3　文件上传处理

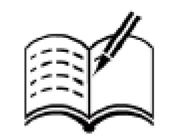
创建文件上传处理页 upfile_chk.php。首先，获取表单提交的数据，定义文件的存储路径。然后，判断是否存在重名文件，判断上传文件大小是否符合要求，生成下载码。接着，完成文件的上传操作，并且将文件信息、下载码和空间使用信息存储到数据库中。最后，输出下载码，返回上一页。其关键代码如下：

```
include_once 'conn/conn.php';
    $filename = $_FILES['upname']['name'];         //读取上传文件名，并保存为数组
    $filetype = $_POST['foundtype'];               //读取上传文件的类别
    $tmpname = $_FILES['upname']['tmp_name'];      //读取临时文件名，并保存为数组
    $tmpsize = $_FILES['upname']['size'];          //上传文件大小
    $tmppub = $_POST['ispub'];                     //是否公开
$dirname=$_SESSION['name'];                    //选取和用户登录名相同的个人文件夹
    $file_path = "upfile/$dirname/";               //文件保存路径
    $max = 0;                                      //文件大小，初始化为 0
for($j=0;$j<count($_FILES['upname']['name']);$j++) {   //判断上传的文件是否已存在
     $filearr=$_FILES['upname']['name'][$j];       //获取上传的文件名
     $filenamei=$file_path.$filearr;               //设置上传文件路径
     if(file_exists($filenamei))                   //判断同目录下文件是否存在
     $construct=0;
     else
     $construct=1;                                 //如果存在则返回 0，不存在则返回 1
    }
if($construct!=1) {                            //判断文件存在，则提示文件名冲突
echo '<script>alert("文件名称冲突！请重新命名后上传！");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';
   }
```


```
if(is_array($tmpsize)){                                  //如果变量$tmpsize 不是数组
        foreach($tmpsize as $value){                     //获取文件总的大小
            $max += $value;
        }
        if($max > 100000000 or $max <= 0){
echo '<script>alert("上传文件总大小大于 100MB，请重新选择");
location="'.$_SERVER['HTTP_REFERE R']. '";</script>';
      }
}else{                                                   //重新选择上传文件
        echo '<script>alert("上传文件错误，请重新检查程序");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';
}
$rightsql = "select * from tb_member where name = '".$_SESSION['name']."'";
        $rightarr = $conne->getRowsRst($rightsql);
        $conne->close_rst();
        switch($rightarr['right']) {                     //判断用户权限
                case '0':
                        $conmax=2000000;         //按当前登录用户的权限分配的最大使用空间
                        break;
                case '1':
                       $conmax=3000000;
                        break;
                case'2':
                       $conmax=5000000;
        }
        echo "<p>"
                $lesarr=$conmax-$rightarr['content'];      //将该用户剩余空间大小赋值给 lesarr 变量
        if($conmax-$rightarr['content'] <= 1000000) { //如果个人剩余空间已经不足 1MB
        echo '<script>alert("您的上传下载空间不足，请清理后上传文件");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';         //提示空间不足
        }
        else if($lesarr-$max<=1000000)                   //如果上传文件大小超过剩余空间大小
                      {
                        echo '<script>alert("您的上传下载空间不足，请整理后上传文件");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';         //提示空间不足
                        }
$chkdownnum = '';
for($i = 0; $i < 15; $i++){
      $chkdownnum .= dechex(rand(0,15));
}
for ($i = 0; $i < count($filename); $i++){               //通过循环将多个文件全部上传
        move_uploaded_file($tmpname[$i],$file—path.$filename[$i]);
        //添加数据库
      $insertsql = 'insert into tb_upfile (filename,filepath,filetype,upauthor, chkdownnum, ispub, filesize)
values("'.trim($filename[$i]).'","'.$file_path.$filename[$i].'","'.$filetype[$i].'",
"'.$_SESSION['name'].'","'.trim($chkdownnum).'",'.$tmppub[$i].','.$tmpsize[$i].')';
        $conne->uidRst($insertsql);
}
$maxtotal=$rightarr['content']+$max;             //从数据库得到该用户已使用空间大小与新上
```





Note

```
传文件占用空间大小的总和
    //更新数据库
    $csql="update tb_member set content = $maxtotal where name= '".$_SESSION['name']."'";
    $conne->uidRst($csql);
    echo '<div style=" padding-top: 30px;">文件上传成功，可以通过下载码：
    <font color="#FF0000">'.$chkdownnum.'</font>  对文件进行下载。</div>';
    echo "<p>点<a href=\"index.php?act=up\"><span style=\"color:red\">这里</span></a>继续上传
其他文件！</p>";
```

说明：

下载码是一组 16 位长的十六进制字符串。先由 rand()函数生成一个 0～15 之间的随机数，再使用 dechex()函数将该数转换为十六进制表示，最后使用 for 循环重复 16 次。如果一次上传多个文件，那么所有文件的注册码为同一个。

3.7　文 件 下 载

3.7.1　功能概述

文件下载是上传、下载模块的另一核心内容。首先，无论用户登录与否，都可以对已公开的文件进行下载；当用户登录后，不但可以上传新的文件，也可以对已经上传的文件执行下载和删除操作；如果当前没有登录，作为游客身份，也可以通过获得与文件匹配的下载码，对不公开文件执行下载操作。文件下载页面的运行效果如图 3.15 所示。

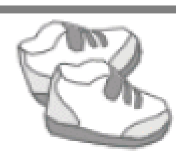


图 3.15　文件下载页面的运行效果

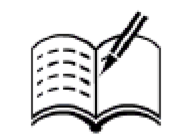
3.7.2　通过下载码下载文件

用户登录后，可以通过单击“下载文件”超链接或输入下载码对文件进行下载。这里以下载码下载为例，讲解文件下载的实现过程。

下载码下载的操作在 index.php 文件中完成。首先，创建下载码文本框，提交下载码。其关键代码如下：


```
<div id="found"> 
下载码：
<input id="chkdown" type="text" /> 
<button id="downbtn" onclick="chkdown()"></button>
```


Note

在 index.php 文件中，当用户单击“下载”按钮时，将触发 onClick()事件，调用 chkdown()函数判断输入值是否为空，如果不为空，将重新加载本页，并将下载码作为参数传递。chkdown()函数存储于 js\index.js 脚本文件中，其关键代码如下。

```
function chkdown(){
    if($('chkdown').value == ''){
        alert('请输入下载码!');
        return false;
    }
    location = '?act=downcode&chkdown='+$('chkdown').value;
}
```

然后，在 index.php 页中，当超链接传递的值等于 downcode 时，加载 downcode.php 页。其关键代码如下：

```
…
<!--根据不同的参数值，载入不同的页面-->
<div id="content">
<?php
    $act = $_GET['act'];
    switch($act){
        case 'up':
            include 'upfile.php';                //载入上传页
            break;
        case 'upfile':
            include 'upfile_chk.php';            //载入上传处理页
            break;
        case 'down':
            include 'downfile.php';              //载入下载页
            break;
        case 'downcode':
            include 'downcode.php';              //载入下载码下载页
            break;
        default:
            include 'pub.php';                   //载入共享文件页
            break;
    }
?>
</div>
```

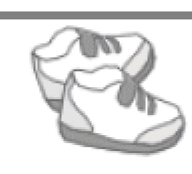
接着，创建 downcode.php 页，根据获取的下载码生成 SQL 查询语句，循环输出下载文件信息；并且创建文件下载超链接，链接到 download.php 文件，完成文件下载操作。其关键代码如下：

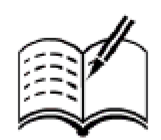
```
<?php
```




```
        $chkdown = trim($_GET['chkdown']);
        $downsql = "select * from tb_upfile where chkdownnum = '".$chkdown."'";
        $downarr = $conne->getRowsArray($downsql);
?>
<!--    下载文件列表    -->
<div id="filelist">
<?php
$infoline = 0;
if(!empty($downarr)){                                           //如果结果集数组不为空
?>
        <ul>
            <li style=" width: 30px; background-color: #F8E7A5;"> </li>
            <li style=" width: 200px; background-color:#F8E7A5; ">文件名</li>
            <li style=" width: 50px; background-color:#F8E7A5; ">文件类型</li>
            <li style=" width: 100px; background-color:#F8E7A5;">下载码</li>
            <li style=" width: 150px; background-color:#F8E7A5;">上传时间</li>
        </ul>
<?php
        foreach($downarr as $key => $value){                    //循环输出数组
?>
        <ul>
            <li style=" width: 30px;background-color: <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
<input id="chk[<?php echo $key; ?>]" type="checkbox" value="<?php echo $value['id']; ?>"
  style=" width:30px; height:20px; border: 0px;" /></li>
            <li style=" width: 200px; background-color:<?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <a href="download.php?id=<?php echo $value['id']; ?>"><?php echo $value['filename']; ?></a></li>
            <li style=" width: 50px; background-color: <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <?php echo $value['filetype']; ?></li>
            <li style=" width: 100px; background-color: <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <?php echo $value['chkdownnum']; ?></li>
            <li style=" width: 150px; background-color: <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <?php echo $value['uptime']; ?></li>
        </ul>
<?php
        $infoline = ($infoline+1)%2;
        }
}else{
        echo '没有可下载资源';
}
?>
</div>
```

最后，创建 download.php 文件，根据文件的完整路径和 header()函数完成下载。其关





键代码如下：

```
<?php
include_once 'conn/conn.php';
	$id = $_GET['id'];                                    //获取用户的 id
	$downsql = "select * from tb_upfile where id = $id"; //通过 where 子句获得相应的文件
	$downarr = $conne->getRowsRst($downsql);
	$path=$downarr['filepath'];                          //获取路径名称赋给$path
		if(!empty($path) and !is_null($path)){
			$filename=basename($path);
			$file=fopen($path,"r");
			header("Content-type:application/octet-stream");
			header("Accept-ranges:bytes");
			header("Accept-length:".filesize($path));
			header("Content-Disposition:attachment;filename=".$filename);
			echo fread($file,filesize($path));
			fclose($file);
			exit;
		}
?>
```

3.7.3　同步删除文件以及释放用户空间

单击“下载文件”超链接后，会在下载文件的下方看到“删除选择的文件”超链接，单击该超链接，在删除数据库文件的同时，也会删除该登录用户相应文件夹下的同名文件，并且同步更新用户使用空间的大小，达到删除文件释放空间的效果。删除文件 1 秒钟之后将返回下载首页。删除文件释放空间的操作是在 downfile.php 文件中完成的，其关键代码如下：

```
<?php
$del = $_GET['del'];
	$num =10;                 //每页显示记录数
	$curpage = $_GET['curpage'];            //当前页
	$sql = "select * from tb_upfile where upauthor = '".$_SESSION['name']."'";
	$totnum = $conne->getRowsNum($sql);                    //记录数
	if($del == 'del'){
		$rd = $_GET['rd'];
		$delsql = "delete from tb_upfile where id = -1 ";
		$deltotal = "select sum(filesize) as total from tb_upfile where id = -1 ";     // 取 得所选文件的总大小
		$content="select * from tb_member where name ='".$_SESSION['name']."'";
		$contarr = $conne->getRowsRst($content);
		if(!empty($rd)){
			$delarr = explode(',',$rd);
			foreach($delarr as $value){
			//以下为新增内容，在删除文件的同时也可以删除硬盘文件夹上的文件
				$sql= "select * from tb_upfile where id = ".$value." ";          //选取删除文件的 id 值
```



```
                $delsql .= " or id = ".$value." ";
            $deltotal .= " or id = ".$value." ";
                $sqlarr = $conne->getRowsRst($sql);              //调用 SQL 函数
                 $delpath=$sqlarr['filepath'];                   //获得删除文件路径
                 unlink($delpath);                               //删除指定路径的文件
            }
            $deltotalnum=$conne->getRowsRst($deltotal);
        $newdata = $contarr['content']-$deltotalnum['total'];
        // $newdata 为删除文件后个人空间的使用大小
         $newsql="update tb_member set content = $newdata where name = '".$_SESSION['name']."'";
        $conne->uidRst($newsql);          //把删除文件后剩余空间大小更新到数据库
        $delnum = $conne->uidRst($delsql);
        echo "<meta http-equiv=\"refresh\" content=\"1;url=index.php?act=down\">";    //1秒钟之后跳转主页
        echo "<p></p>";
        echo "删除成功！ ";
            }
```

3.8　后台管理

3.8.1　后台管理概述

在后台管理系统中只有一个功能，就是管理注册用户的权限。其运行效果如图 3.16 所示。

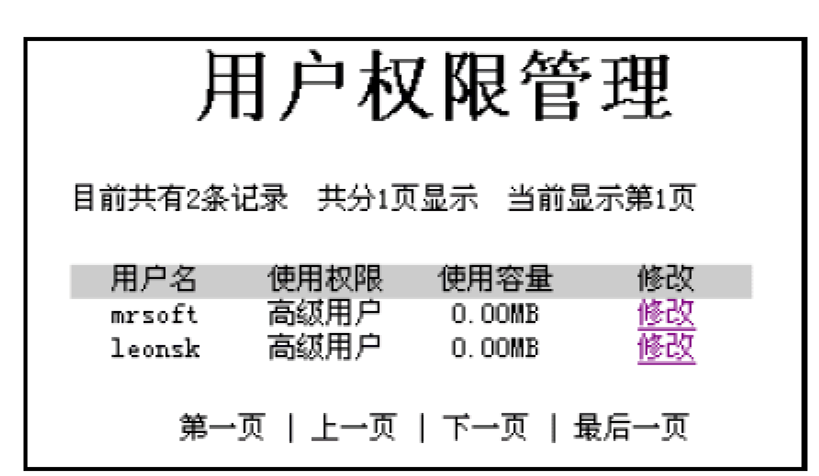


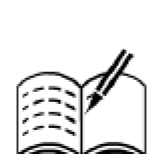
图 3.16　用户权限管理

3.8.2　用户权限管理页面设计

进入到后台管理系统中，以分页形式输出已经注册的用户信息，包括用户名、使用权限、使用容量，以及“修改”超链接。admin.php 文件的关键代码如下：

```
<?php
include_once 'conn/conn.php';
$list_num=10;
$sql = "select * from tb_member";                        //查询语句
$sqlarr = $conne->getRowsRst($sql);                      //返回查询结果集
$totnum = $conne->getRowsNum($sql);                      //获取查询记录数
$conne->close_rst();
if($totnum>0){
```





Note

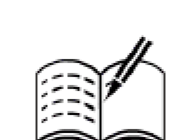
```
?>
<p>
<table>
<tr class="tr1">
<td>用户名</td>
<td>使用权限</td>
<td>使用容量</td>
<td>修改</td>
</tr>
<?php
$sql = $sql." order by id desc limit ".($list_num *($page-1)).",".$list_num;   //查询语句
$filearray = $conne->getRowsArray($sql);
foreach($filearray as $value){
echo "<tr>\n";
echo "<td>".$value['name']."</td>\n";
switch($value['right']){                                                    //查看用户权限
        case '0':
                echo "<td>普通用户</td>\n";
                 break;
        case '1':
                echo "<td>会员用户</td>\n";
                  break;
        case '2':
                echo "<td>高级用户</td>\n";
                 break;
        }
        echo "<td>".sprintf("%-01.2f",($value['content']/1000000))."MB</td>\n";
        echo "<td><a href=changesub.php?id=".$value['id']."">修改</a></td>\n";     //获取 id 更改用户权限
        echo "</tr>\n";
        }
echo "</table>";
```

3.8.3 更改用户权限表单设计

更改用户权限，是从用户权限管理页中设置的“修改”超链接开始的。在 changesub.php 文件中，根据超链接传递的 ID 值，查询出指定用户的详细信息，并且将该信息在表单中输出，进而通过表单将数据提交到指定页，完成对用户权限的修改操作。changesub.php 文件的关键代码如下：

```
<table border=1>
<form action="change.php" method="post">
<?php
include_once 'conn/conn.php';
$sql="select * from tb_member where id = $id";                          //查询语句
$changearr = $conne->getRowsRst($sql);                                  //返回查询结果
?>
<tr>
```





Note

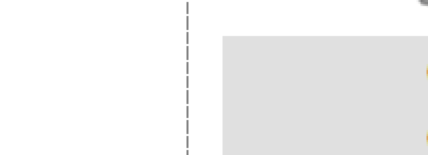
```
<td>用户名</td>
<td><?php echo $changearr['name'];?></td>
</tr>
<tr>
<td>当前权限</td>
<td>
<?php
switch($changearr['right']){                                    //获取用户权限
        case '0':
          echo "普通用户";
        break;
        case '1':
          echo "会员用户";
          break;
        case '2':
          echo "高级用户";
         break;
        }
        ?>
</td>
</tr>
<tr>
<td>修改权限</td>
<td>
<select name="right">                                           //选择用户权限
<option value="0" selected="selected">普通用户</option>
<option value="1">会员用户</option>
<option value="2">高级用户</option>
</select>
</td>
</tr>
<tr><td colspan=2>
<center>
<input type="submit" value="提交" />
</center>
</td>
</tr>
<input type="hidden" name="id" value="<?php echo $changearr['id']?>" />      //添加一个隐藏域传递 id 参数
</form>
</table>
```

3.8.4 执行权限更改操作

在 changesub.php 文件中创建指定的权限更改处理文件 change.php，获取表单中提交的数据，完成用户权限的更改操作。change.php 文件的关键代码如下：

```
<?php
include_once 'conn/conn.php';
```






```
$right=$_POST['right'];                                   //获得权限更改表单的传递值
$id=$_POST['id'];
$upsql = "update tb_member set tb_member.right =$right where id = $id";   //查询语句
$num=$conne->uidRst($upsql);                                              //返回结果
$conne->close_rst();
echo "<p>";
if($num==1){
    echo "权限更改成功！ ";
    echo "<p>";
}else{
    echo "权限更改失败！ ";
    echo "<p>";
}
?>
<!--以下为 html 内容，修改成功都会返回修改主页--!>
<meta http-equiv="refresh" content="2;url=admin.php" />
2seconds    later to return the manger page!
```

3.9 技术提炼

3.9.1 php.ini 配置文件

如果要在 PHP 中实现小文件的上传（2MB 以下），那么无须对 php.ini 配置文件进行修改，使用默认参数即可。但如果想实现完美的上传功能，则一定要对配置文件有所了解。在 php.ini 文件中，与文件上传有关的设置如图 3.17 所示。

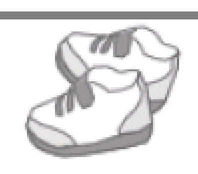
```
239
240 ;;;;;;;;;;;;;;;;;;;
241 ; Resource Limits ;
242 ;;;;;;;;;;;;;;;;;;;
243
244 ; Maximum execution time of each script, in seconds
245 max_execution_time = -1
246 ; Maximum amount of time each script may spend parsing request data
247 max_input_time = -1
248 ; Maximum amount of memory a script may consume (16MB)
249 memory_limit = 32M
250

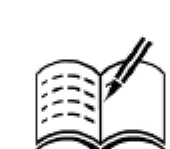
410
411 ; Maximum size of POST data that PHP will accept.
412 post_max_size = 16M
413

504
505 ;;;;;;;;;;;;;;;;
506 ; File Uploads ;
507 ;;;;;;;;;;;;;;;;
508
509 ; Whether to allow HTTP file uploads.
510 file_uploads = On
511
512 ; Temporary directory for HTTP uploaded files (will use system defau
513 ; specified).
514 upload_tmp_dir = "D:\xampp\tmp"
515
516 ; Maximum allowed size for uploaded files.
517 upload_max_filesize = 32M
518
```

图 3.17　php.ini 文件中的相关参数

从图 3.17 中可以看到，有如下 3 部分内容需要进行配置。





Note

1. Resource Limits

Resource Limits，直译过来就是资源限制，其中包含 3 个参数。该区块不仅仅是针对上传、下载的，而是对全部的文件进行设置。各个参数含义及参数值说明如表 3.1 所示。

表 3.1　Resource Limits 块的参数说明

参　数	描　述
max_execution_time	每个脚本页面完成执行的最大时间，单位是秒。如果设为-1，说明没有限制
max_input_time	每个脚本页面处理请求数据的最大时间，单位是秒，也可以设为-1
memory_limit	一个脚本页所能够消耗的最大内存

2. post_max_size

post_max_size 参数指 PHP 通过表单 POST 所能接收的最大值，包括表单里的所有项。

3. File Uploads

File Uploads 块是专为文件上传设置的。该区块中同样包含了 3 个参数，各个参数含义及参数值说明如表 3.2 所示。

表 3.2　File Uploads 块的参数说明

参　数	描　述
file_uploads	是否允许HTTP上传，默认为On，即开启，无须修改
upload_tmp_dir	文件上传时的临时存储目录。如果没指定，将会用系统默认的临时文件夹
upload_max_filesize	允许上传的文件最大值

3.9.2　上传表单设置

配置文件修改完毕后，重启 Apache 服务器。接下来，开始设计<form>表单。<form>表单可以实现人机交互，是动态网页的一个最基本功能。最初的<form>表单并不支持文件上传，表单中 input 元素的 type 属性只有 7 种值，即文本框（text）、密码框（text）、单选框（radio）、多选框（checkbox）、重置按钮（reset）、确定按钮（submit）和图像（image）。后来，rfc1867 为 HTTP 协议添加了这个功能，目前几乎所有的浏览器和服务器都支持此协议。

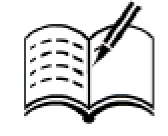
rfc1867 协议为 type 属性增加了 file 值，同时，<form>标签中的 ENCTYPE 属性值必须设为 multipart/form-data，method 属性则必须是 POST，3 个条件缺一不可。当表单属性设置正确后，才可以在表单处理页中获取到上传文件的值。下面是简单、可用的上传表单代码：

```
<form method="post" action="2.php" enctype="multipart/form-data">
        <input name="filename" type="file" />
        <input type="submit" value="enter"/>
</form>
```

此外，还建议在上传表单中增加一个 hidden 隐藏域，隐藏域的 name 值为 MAX_FILE_SIZE，value 值是允许上传的最大字节数。例如：

```
<input type="hidden" name="MAX_FILE_SIZE" value="1024000" />
```

这样，当上传文件大于 1MB 时，$_FILES 中的 error 值将返回 2。关于$_FILES 变量的使用在 3.9.3 节中给出。该隐藏域的作用不是真正要去限制上传文件的大小，因为要绕过该项检查非常简单，其真正作用是为了避免用户误传了大文件而陷入“无尽地”等待之中。


Note

注意：

通过隐藏域控制上传文件的大小，必须将隐藏域放置在文件域之前，否则隐藏域不会生效。

3.9.3　预定义变量$_FILES

如果表单设计得没有问题，那么在处理页中，使用$_FILES 预定义变量就可以得到上传文件的相关信息；反之，如果得不到，那么就有可能是配置文件或表单设置的问题了。下面先介绍预定义变量$_FILES。

$_FILES 变量存储的是上传文件的相关信息，这些信息对于上传功能有很大作用。该变量是一个二维数组，保存的信息如表 3.3 所示。

表 3.3　$_FILES 变量值

参　数	描　述
$_FILES[filename]['name']	存储了上传文件的文件名，如exam.txt、myDream.jpg等
$_FILES[filename]['size']	存储了文件大小，单位为字节
$_FILES[filename]['tmp_name']	文件上传时，首先在临时目录中被保存成一个临时文件。该变量为文件名
$_FILES[filename]['tmp_name']	上传文件的类型。注意是类型，不是后缀
$_FILES[filename]['type']	存储了上传文件的错误代码

error 字段将返回上传文件的错误代码，一共有 7 种，具体说明如表 3.4 所示。

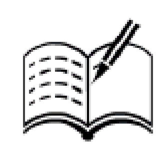
表 3.4　$_FILES[filename]['eror']中的错误代码及说明

错 误 代 码	错 误 说 明
0	UPLOAD_ERR_OK：上传成功，无错误
1	UPLOAD_ERR_INI_SIZE：文件大小超出php.ini中的upload_max_filesize的参数值
2	UPLOAD_ERR_FORM_SIZE：文件大小超出表单中MAX_FILE_SIZE隐藏域的值
3	UPLOAD_ERR_PARTIAL：文件没有被全部上传
4	UPLOAD_ERR_NO_FILE：没有文件被上传
6	UPLOAD_ERR_NO_TMP_DIR：找不到临时文件目录
7	UPLOAD_ERR_CANT_WRITE：文件写入失败

下面使用 3.6.2 节中的表单上传一个文件，使用$_FILES 变量输出文件信息。代码如下：

```
<?php
    $tmparr = $_FILES['filename'];                          //获取文件域的值
    if($tmparr['name'] != ''){                              //如果上传文件名不为空
        foreach($tmparr as $name => $value){
```






```
                echo $name.' => '.$value.'<br>';            //使用循环输出数组索引和值
            }
        }else{
            echo "<script>alert('请添加上传文件');history.go(-1);</script>";
        }
?>
```

运行结果如图 3.18 所示。

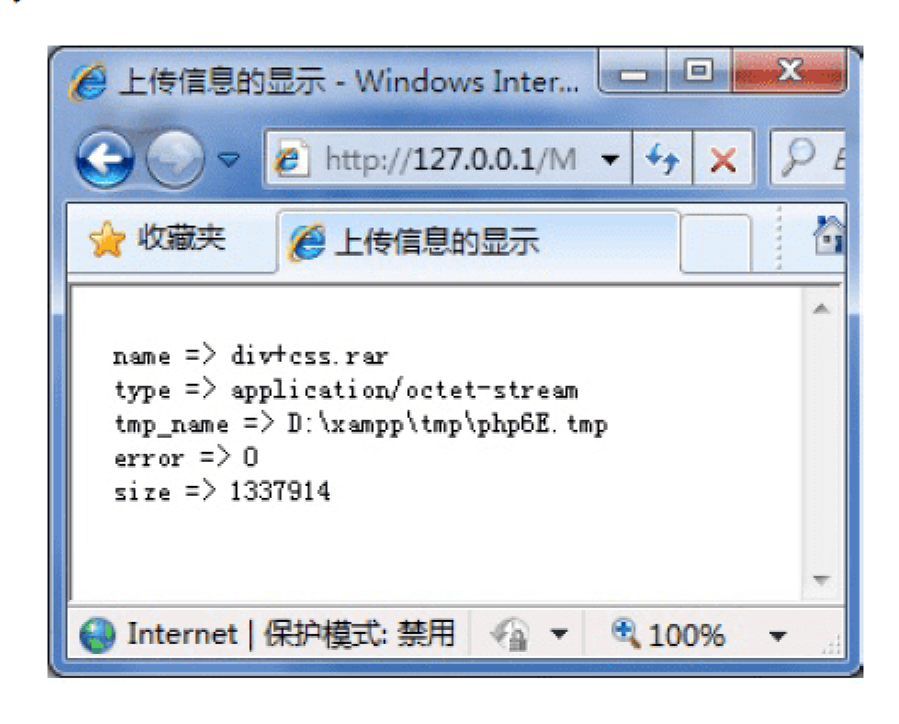


图 3.18　获取上传文件的信息

3.9.4　上传文件函数

move_uploaded_file()函数是 PHP 中专门用来上传文件的。其语法格式如下：

```
bool move_uploaded_file ( string filename, string destination )
```

☑ 功能：上传文件到指定的位置。如果成功返回 True，否则返回 False。

☑ 参数说明：有两个参数，分别介绍如下。

- ✧ filename：上传文件的临时文件名，即$_FILES['tmp_name']。
- ✧ destination：上传后保存的新的路径和名称。

通过 3.6.3 节和 3.7.3 节的介绍，细心的读者可能已经发现，真正的上传文件已经被保存到临时目录里，move_uploaded_file()函数的作用只是将文件换个地方保存，这是出于安全的考虑。

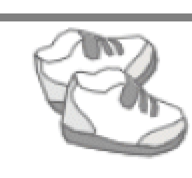
此外，还可以通过 is_uploaded_file()函数判断文件是否为 POST 上传。其语法格式如下：

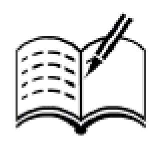
```
bool is_uploaded_file ( string filename )
```

3.9.5　多文件上传

多文件上传和单文件上传没什么区别，只要将表单中的文件域用数组命名即可。代码如下：

```
<form action="2.php" method="post" enctype="multipart/form-data">
   <input name="userfile[]" type="file" /><br />
   <input name="userfile[]" type="file" /><br />
      ……………
   <input type="submit" value="上传" />
</form>
```





Note

上传的多个文件可以通过 for 循环语句进行获取。其方法如下：

```
$filename= $_FILES['userfile']['name'];
for ($i = 0; $i < count($filename); $i++){              //通过循环将多个文件全部上传
        move_uploaded_file($tmpname[$i],$file_path.$filename[$i]);
        //添加数据库
        $insertsql = 'insert into tb_upfile (filename,filepath,filetype,upauthor,chkdownnum,ispub)
values("'.trim($filename[$i]).'","'.$file_path.$filename[$i].'","'.$filetype[$i].'",
"'.$_SESSION['name'].'","'.trim($chkdownnum).'",'.$tmppub[$i].')';
        $conne->uidRst($insertsql);
}
echo '<div style=" padding-top: 30px;">文件上传成功
<font color="#FF0000">'.$chkdownnum.'</font> 对文件进行下载。</div>';
```

说明：

本模块中并没有加入多文件上传功能，因为多文件上传存在一些弊端，如在上传多个文件的时候，多个文件会获取相同的下载码；在上传多个文件的时候，判断多个文件上传是否有重名比较麻烦，一旦上传失败，会导致所有文件都上传失败，必须重新上传。这时可以在上传页面中增加 JavaScript 脚本，在提交之前判断是否存在同名文件。如果不加入这个判断过程，在其后删除多个同名文件的时候就会导致出错。有兴趣的读者可以自行研究一下。

3.9.6 文件下载

在一个完整的网站中，拥有文件下载功能能够给网站带来更多的访问者，提高网站的访问量。实现下载最常用的方式有通过链接下载和通过 header()函数下载两种。

（1）通过链接下载

通过链接下载，就是将文件的相对路径或绝对路径直接作为超链接来下载。如果是压缩文件，那么直接单击超链接就可以下载；对于文本文件，单击鼠标右键，在弹出的快捷菜单中选择“目标另存为”命令，就可以将该文件下载下来。所有格式的文件，都可以通过这种方法下载，运行结果如图 3.19 所示。

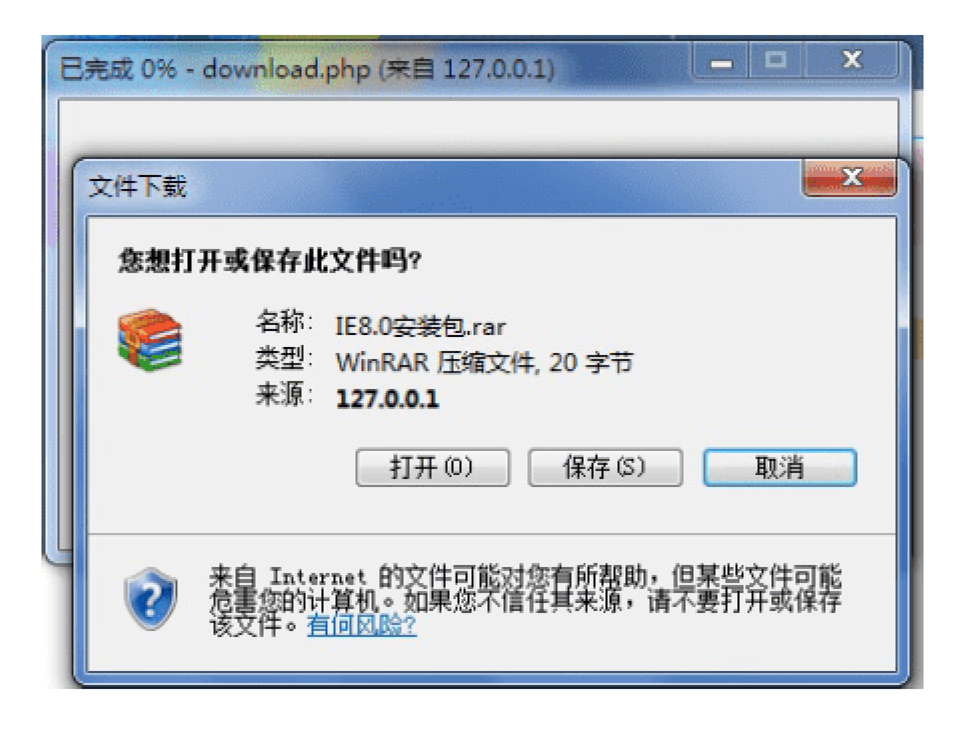
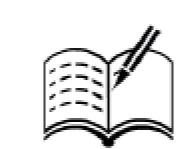


图 3.19　通过链接下载

（2）通过 header()函数下载





header()函数属于 HTTP 函数，其作用是发送一个原始的 HTTP 标头。其语法格式如下：

```
void header ( string string [, bool replace [, int http_response_code]] )
```

参数说明如下。

- string：发送的标头。
- replace：如果一次发送多个标头，指明相似的标头是替换还是添加。如果是 False，则强制发送多个同类型标头；默认是 True，即替换。
- http_response_code：强制 HTTP 响应为指定值。

HTTP 标头有很多，这里只介绍下载，其他参见相关资料。下载 HTTP 标头的代码如下：

```
header('Content-Disposition: attachment; filename="filename"');
```

上面的代码中唯一需要改动的就是 filename，将 filename 替换为要下载的文件就可以完成下载。

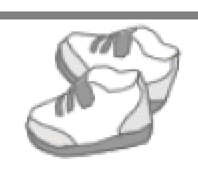


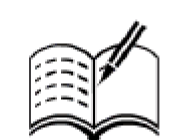
网页计数器模块

（自学视频、源程序：配套资源\mr\4\）

在浏览网页时经常可以看到网站的首页带有一个制作精美的计数器，用来显示该网站的访问量。它不仅可以美化网站的页面，还能吸引更多浏览者的注意力，提高网站的访问量。网页计数器的形式多种多样，有直接以数字计数存储于文本或数据库中的，也有以图形方式计数存储于文档或数据库中的……通过本章的学习，读者能够学到以下内容。

- 简单数字计数器
- 图形数字计数器
- 利用 SESSION 实现无刷新计数器
- GD2 图形计数器
- 数据库数字计数器
- Cookie 计数器





4.1　网页计数器模块概述

4.1.1　模块概述

网页计数器对于网站管理者来说是一个非常值得关注的部分，它记录了网站被访问的次数，客观地反映了网站受欢迎的程度。如果网站的点击率很高，访问者也会认为网站值得关注，因而耐心驻足，进而增加网站的访问量，提高网站的知名度。

4.1.2　业务流程图

网页计数器模块主要从计数器功能实现的方法着手，详细地介绍了简单数字计数器、图形数字计数器、GD2 图形数字计数器、数据库数字计数器的实现方法。

网页计数器的操作流程如图 4.1 所示。

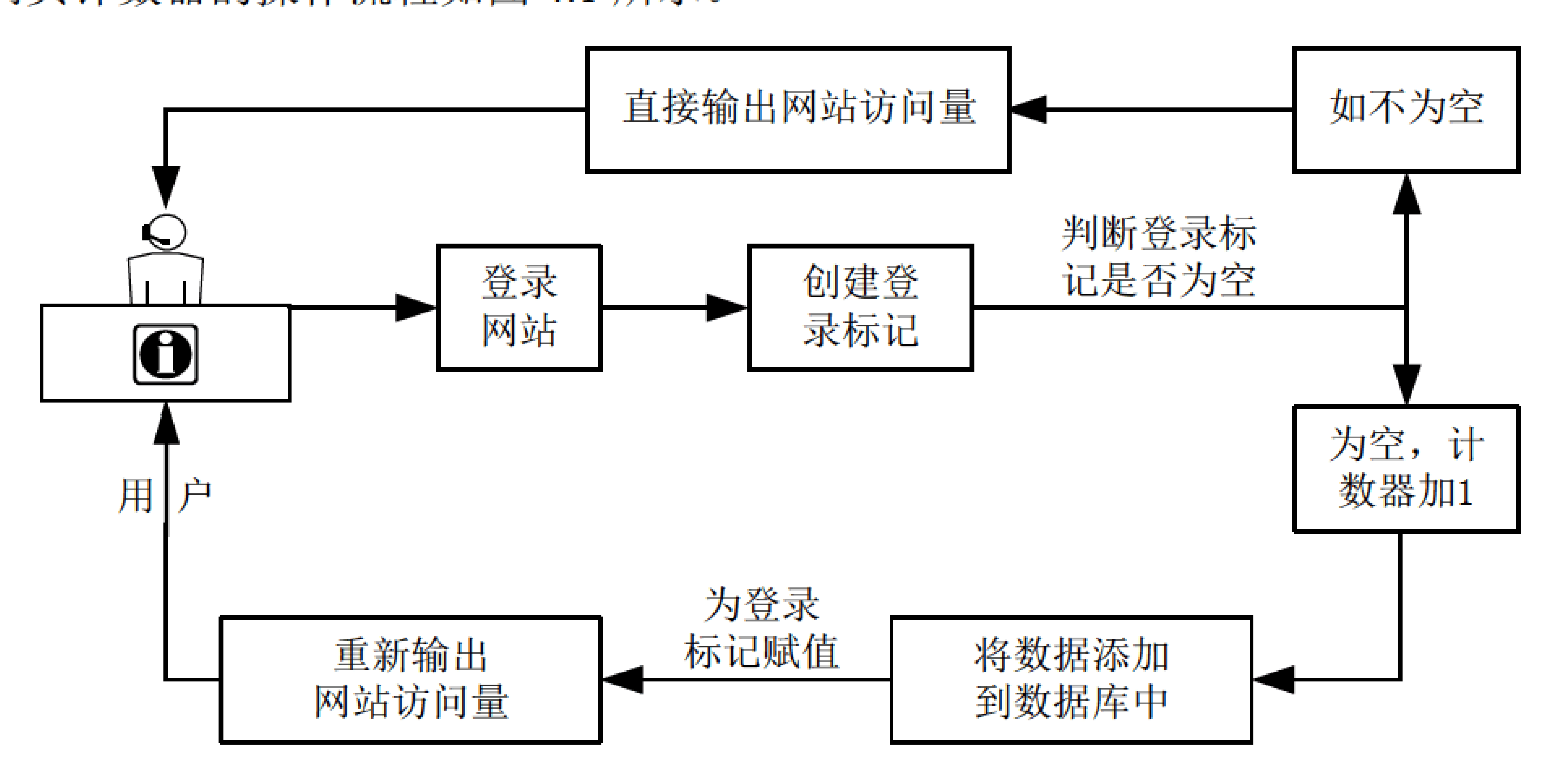


图 4.1　网页计数器的操作流程

4.2　简单数字计数器

数字计数器是最简单的一种，它将数据存储在文本文件中。其思路是首先判断文本文件是否存在，如不存在则打开失败，如打开成功则继续执行并读取文件中的数据，将计数器的值加 1；然后以写的方式重新打开文件，把新的统计数据写入到文件中，关闭文件；最后重新打开文件，读取并输出文件中的数据。

简单数字计数器的操作流程如图 4.2 所示。



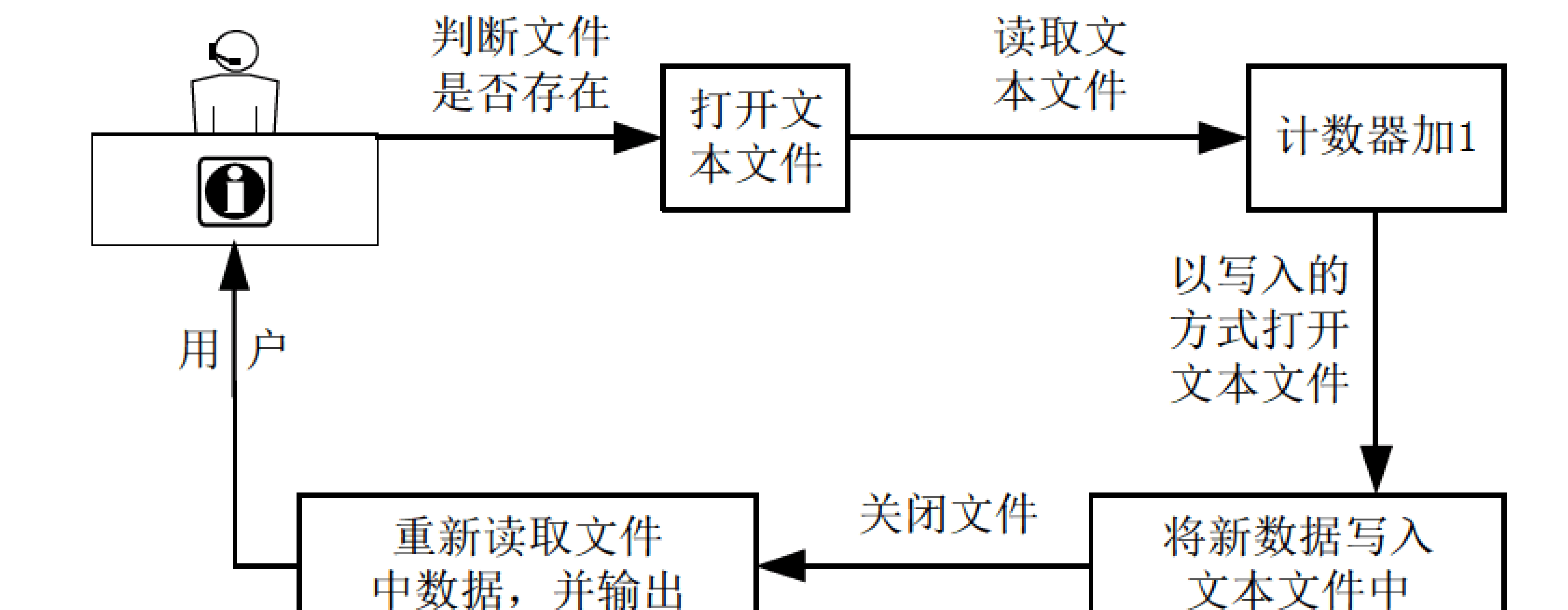



图 4.2　简单数字计数器的操作流程

按上述原理设计的计数器，当页面刷新时其值就会增加。因此，要想控制页面刷新时计数器的值不增加，最好是应用 SESSION 变量防止重复计数。

当应用 SESSION 防止重复计数后，无论如何刷新页面计数器的值都不会增加。只有在重新打开页面时，计数器的值才会继续增加。简单数字计数器的运行效果如图 4.3 所示。

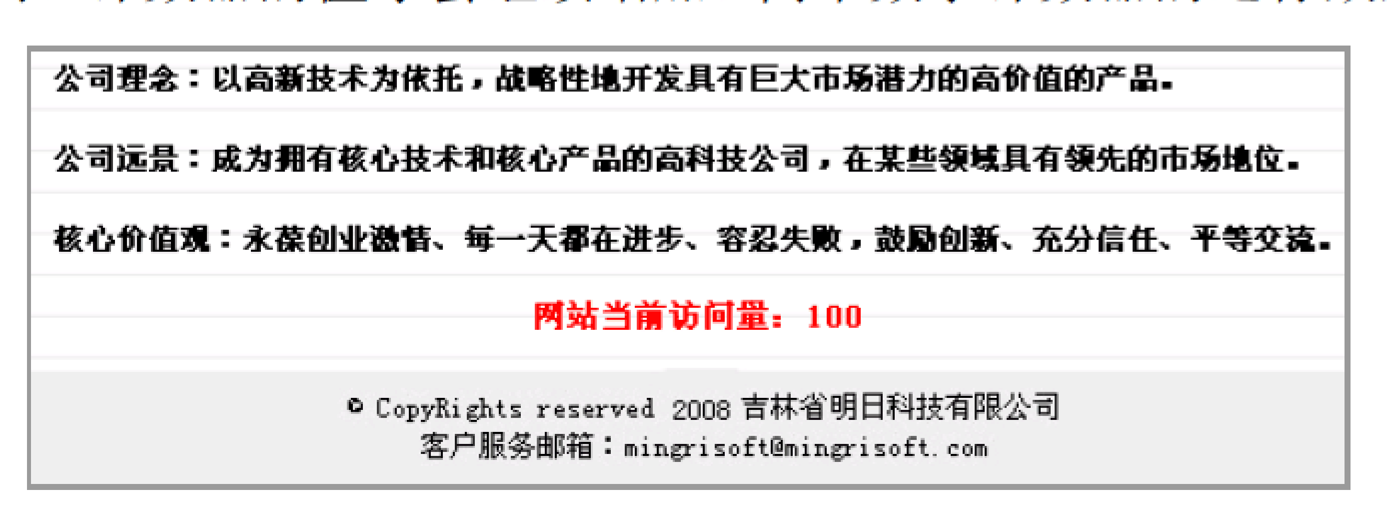


图 4.3　简单数字计数器

下面对简单数字计数器进行详细分析。

首先通过 session_start()函数初始化一个 SESSION 变量，通过该变量来控制页面刷新对计数器的影响。

然后，通过 if 语句判断 SESSION 变量的值是否为空，如果为空，则打开指定的文本文件，否则将不执行。

如果 SESSION 变量的值为空，则调用 fopen()函数以只读的方式打开指定的文本文件，如果文件不存在则输出打开文件失败，否则将继续执行，读取文件中数据。这里将指定的文本文件 counter.txt 存储在根目录下。

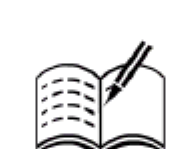
如果文本文件存在，则继续执行，将调用 fgets()函数读取文本文件中的数据，并通过 fclose()函数关闭文件。

此时将读取到的数据$counter 加 1，即将计数器的值加 1。通过 fopen()函数重新打开指定的文本文件（注意，是以写入的方式打开），并调用 fputs()函数将新统计的数据写入到文本文件中，关闭文件。

最后为 SESSION 变量赋值，将 SESSION 变量赋值为 1，用于控制因网页刷新而导致计数器的值无限增加。

简单数字计数器的文件存储于 mr\4\01\index.php 文件中，其关键代码如下。





Note

```
<?php
//使用文本文件存储数据
if(!isset($_SESSION['temp'])){   //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义变量
        if(($fp=fopen("counter.txt","r"))==false){
                echo "打开文件失败!";
        }else{
                $counter=fgets($fp,1024);                //读取文件中数据
                fclose($fp);                             //关闭文本文件
                $counter++;                              //计数器的值加 1
                $fp=fopen("counter.txt","w");            //以写的方式打开文本文件
                fputs($fp,$counter);                     //将新的统计数据加 1
                fclose($fp);
        }                                                //关闭文件
                $_SESSION['temp']=1;        //计数器的值增加后，为$_SESSION[temp]赋值 1
}
//从文本文件中读取统计数据
if(($fp=fopen("counter.txt","r"))==false){
        echo "打开文件失败!";
}else{
        $counter=fgets($fp,1024);
        fclose($fp);
}
?>
```

（1）fopen()函数

该函数用于打开某文件，并返回该文件的标识指针。其语法格式如下：

```
resource fopen(string filename, string mode [, int use_include_path [, resource context]])
```

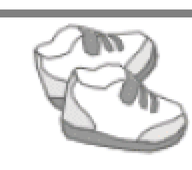
其中参数说明如表 4.1 所示。

表 4.1　fopen()函数的参数说明

参　　数	说　　明
filename	必要参数。用于指定要打开文件的本地或远程地址
mode	必要参数。用于指定打开文件的模式
use_include_path	可选参数。如果将该参数设置为true，PHP会尝试按照include_path标准包含路径中的每个指向去打开文件
context	可选参数。设置提高文件流性能的一些选项

在 fopen()函数的 mode 参数中，其指定的打开模式很多，其中常用的有如下几种。

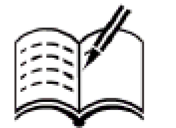
- ☑ 'r'：只读方式打开，将文件指针指向文件头。
- ☑ 'r+'：读写方式打开，将文件指针指向文件头。
- ☑ 'w'：写入方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在，则尝试创建。
- ☑ 'w+'：读写方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在，则尝试创建。



（2）fgets()函数

该函数用于获取文件指针所在行指定长度的内容。其语法格式如下：

```
string fgets(int handle [,int length])
```



Note

参数说明如下：

☑ handle：必要参数，指调用 fopen()或 fsockopen()函数成功打开文件或连接上服务器后所返回的文件标识。

☑ length：可选参数，用于指定读取一行最多的字节数，并返回 length-1 字节的字符串。如果省略该参数，则 length 的长度默认为 1024 字节。



（3）fputs()函数

fwrite()函数的别名，用于将字符串写入到指定的文件中，并可以规定写入字节的大小。其语法格式如下：

```
int fwrite( resource handle, string string [, int length])
```

其中参数说明如表 4.2 所示。

表 4.2 fputs()函数的参数说明

参　数	说　明
handle	必要参数。文件标识指针
string	必要参数。要写入某文件的字符串
length	可省参数。指写入文件的长度，如果省略该参数则将指定字符串的所有内容写入到文件中

至此，简单数字计数器的计数部分设计完成，接下来要做的就是在网页中重新输出文本文件中的数据，即输出网站当前的访问量。

```
<?php
//从文本文件中读取统计数据
	if(($fp=fopen("counter.txt","r"))==false){		//打开指定文件
		echo "打开文件失败!";
	}else{
		$counter=fgets($fp,1024);			//读取文件内容
		fclose($fp);
		echo "数字计数器: " .$counter ;		//输出访问次数
	}
?>
```

4.3　图形数字计数器

所谓图形数字计数器，就是将计数器中的数据以数字图形的形式显示出来。其设计原理与简单数字计数器类似，唯一区别的地方就是在对数据的输出过程中，数字计数器直接将数据进行输出，而图形数字计数器则将数据转化成数字图形的形式进行输出。其运行效果如图 4.4 所示。





图 4.4　图形数字计数器



计数器功能的实现方法已经在简单数字计数器中进行了详细的讲解，这里不再赘述。下面主要讲解如何将文本文件中的数据以图形的形式进行输出。

图形数字计数器存储于 mr\4\02\文件夹下，其实现关键步骤如下。

首先，读取文本文件中的数据，将其写入到一个变量中。

```
<?php session_start();
if(!isset($_SESSION['temp'])){          //判断$_SESSION[temp]==""的值是否为空，其中的temp 为自定义变量
    if(($fp=fopen("counter.txt","r"))==false){
        echo "打开文件失败!";
    }else{
        $counter=fgets($fp,1024);       //读取文件中数据
        fclose($fp);                    //关闭文本文件
        $counter++;                     //计数器的值加 1
        $fp=fopen("counter.txt","w");   //以写的方式打开文本文件
        fputs($fp,$counter);            //将新的统计数据加 1
        fclose($fp);
    }                                   //关闭文件
    $_SESSION['temp']=1;                //为$_SESSION[temp]赋一个值
}
?>
```

然后，将读取的数据以图形的形式进行输出。这里设计的是一个 6 位数字的图形计数器，将 0～9 的数字图形存储在根目录下的 images 文件夹中。步骤如下：

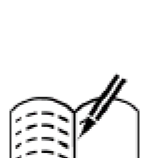
（1）通过 strlen()函数获取数据的长度。

（2）输出 6 位数字的图形计数器中的空值，即处理 6 位数字中补位值 0 的输出。首先应用 str_repeat()函数获取 6 位数字中补位值 0 的个数；然后应用 for 语句，根据补位值 0 的个数（$str），循环输出代表 0 的图形。

（3）将文本文件中存储的数据以图形的形式输出。应用 for 循环语句，根据 strlen()函数获取变量$counter 的长度，循环输出指定的图形；应用 switch 语句为数据指定不同的图形。

```
$len=strlen($counter);                      //获取字符串的长度
$str=str_repeat("0",6-$len);                //获取 6-$len 个数字 0
for($i=0;$i<strlen($str);$i++){             //获取变量$str 的字符串长度
    $result=$str[$i];
    $result='<img src=images/0.gif>';
    echo $result;                           //循环输出$result 的结果
```






```
        }
        for($i=0;$i<strlen($counter);$i++){          //获取字符串的长度
            $result=$counter[$i];                    //获取数据
            switch($result){                         //为不同的数据指定不同的图形
                //如果值为"0"，则输出 0.gif 图片
            case "0"; $ret[$i]="0.gif";    break;
                case "1"; $ret[$i]="1.gif";    break;
                case "2"; $ret[$i]="4.gif";    break;
                case "3"; $ret[$i]="3.gif";    break;
                case "4"; $ret[$i]="4.gif";    break;
                case "5"; $ret[$i]="5.gif";    break;
                case "6"; $ret[$i]="6.gif";    break;
                case "7"; $ret[$i]="7.gif";    break;
                case "8"; $ret[$i]="4.gif";    break;
                case "9"; $ret[$i]="9.gif";    break;
            }
            echo "<img src=images/".$ret[$i].">";       //输出文本文件中存储的数据
        }
    }
    ?>
```

说明：

（1）strlen()函数：获取指定字符串的长度。

（2）str_repeat()函数：将指定的字符串重复输出。

4.4　GD2 图形计数器

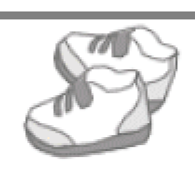
采用 GD2 图形计数器，可以将计数器中存储的数据以 GD2 函数绘制的图形的形式进行输出，如图 4.5 所示。



图 4.5　GD2 图形计数器

计数器中的数据仍然存储在文本文件中，至于如何存储和读取其中的数据这里不再赘述，直接讲解如何将读取的数据以 GD2 图形的形式进行输出。GD2 图形计数器存储于 mr\4\03 文件夹下，其关键实现步骤如下。

（1）创建一个画布，并指定画布背景的颜色和字体的颜色。首先通过 imagecreate()函数创建画布，然后调用 imagecolorallocate()函数来分别定义背景和字体的颜色（其中在定义字体的颜色时用到了 rand()函数，通过其在指定的范围内随机选择颜色）。



（2）应用 iconv()函数对指定的中文字符进行转换，并应用 imagettftext()函数输出中文字符串。

（3）应用 imagestring()函数输出计数器中存储的数据，并应用 imagepng()函数以 PNG 格式将图像输出到浏览器或文件中，通过 imagedestroy()函数销毁图像。



```
<?php
if(($fp=fopen("counter.txt","r"))==false){
        echo "打开文件失败!";
}else{
        $counter=fgets($fp,1024);
        fclose($fp);
       //通过 GD2 函数创建画布
        $im=imagecreate(240,24);
        $gray=imagecolorallocate($im,255,255,255); //定义背景
        $color =imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255));    //定义字体颜色
        //输出中文字符
        $text="couter:";                          //对指定的中文字符串进行转换
        //输出网站的访问次数
        imagestring($im,5,160,5,$text.$counter,$color);
        imagepng($im);
        imagedestroy($im);                        //以 PNG 格式将图像输出到浏览器
}
?>
```

（1）imagecreate()函数

该函数用于创建一个新的图像，返回一个图像标识符。其语法格式如下：

```
resource imagecreate ( int x_size, int y_size )
```

参数 x_size、y_size 为图像的尺寸，单位为像素（pixel）。

（2）imagecolorallocate()函数

该函数将返回一个标识符，代表了由给定的 RGB 成分组成的颜色。其语法格式如下：

```
int imagecolorallocate ( resource image, int red, int green, int blue)
```

参数说明如下。

☑ image：imagecreatetruecolor()函数的返回值。

☑ red、green 和 blue：分别是所需要颜色的红、绿、蓝成分，其值是 0～255 的整数或者十六进制的 0x00～0xFF。

imagecolorallocate()必须被调用以创建每一种用在 image 所代表的图像中的颜色。第一次对 imagecolorallocate()函数的调用会填充背景色。

（3）rand()函数

该函数用于产生一个随机数，并返回随机数的值。其语法格式如下：

```
int rand ( int min, int max)
```

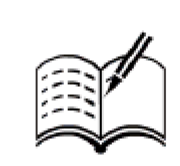
如果没有提供可选参数 min 和 max，则返回 0～RAND_MAX 之间的伪随机数；如果



提供可选参数，则返回可选参数范围之内的随机数。

（4）iconv ()函数

该函数用于将一种已知的字符集文件转换成另一种已知的字符集文件。其语法格式如下：



Note

```
iconv ( string in_charset, string out_charset, string str)
```

（5）imagettftext ()函数

该函数用于以 TTF（TrueType Fonts）字体向图像中绘制字符串。其语法格式如下：

```
array imagettftext ( resource image, float size, float angle, int x, int y, int color, string fontfile, string text )
```

其中参数说明如表 4.4 所示。

表 4.4　imagettftext()函数的参数说明

参　数	说　明
image	图像资源
size	字体大小。根据GD版本不同，应该以像素大小（GD1）或点大小（GD2）指定
angle	字体的角度，顺时针计算，0度为水平，也就是3点钟的方向（由左到右），90度则为由下到上的文字
x	文字的x坐标。设定第一个字符的基本点
y	文字的y坐标。设定字体基线的位置，不是字符的最底端
color	文字的颜色
fontfile	字体的文件名称，也可以是远端的文件
text	字符串内容

（6）imagestring()函数

利用该函数可实现水平地绘制一行字符串。其语法格式如下：

```
bool imagestring ( int im, int font, int x, int y, string str, int col )
```

imagestring()函数用 col 颜色将字符串 str 绘制到 im 所代表的图像的 x,y 坐标处（这是字符串左上角坐标，整幅图像的左上角为 0,0）。如果 font 是 1、2、3、4 或 5，则使用内置字体。

至此，GD2 图形计数器讲解完毕。上述介绍的这 3 种计数器都是从输出的方式上进行区别的，其统计的方式是完全相同的，下面将要继续介绍几种完全不同的计数器。

4.5　数据库数字计数器

顾名思义数据库数字计数器，就是将统计的数据存储到数据库中。相对于其他几种计数器而言，数据库数字计数器的功能是最为强大的，不但可以统计出一个数值，还可以记录更多的信息（如 IP 地址、访问时间等），这些都可以存储在数据库中。其运行效果如图 4.6 所示。



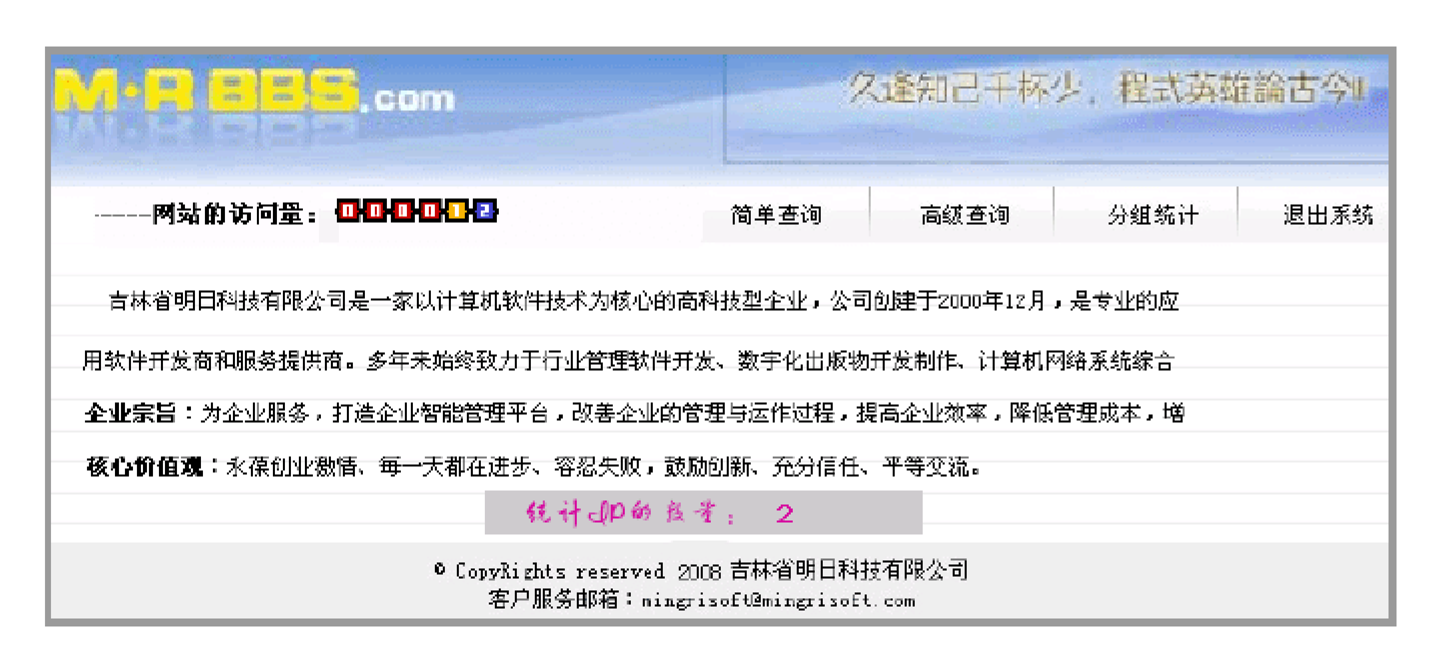


图 4.6　数据库数字计数器

既然是数据库数字计数器，就需要数据库的支持。首先要创建一个数据库，并在其中创建一个数据表用于存储数据。创建的数据表如图 4.7 所示。



字段	类型	整理	属性	Null	默认	额外	操作
id	int(10)			否		auto_increment	
counts	varchar(50)	gb2312_chinese_ci		否			
data1	varchar(50)	gb2312_chinese_ci		否			
data2	varchar(50)	gb2312_chinese_ci		否			
ip	varchar(50)	gb2312_chinese_ci		否			

图 4.7　创建的数据表

然后连接数据库。连接数据库文件 mr\4\04\conn\conn.php 的代码如下：

```
<?php
    $id=mysql_connect('localhost','root','111');          //连接数据库服务器
    mysql_select_db('db_counter',$id);                     //连接 db_counter
    mysql_query("set names gb2312");                       //指定数据库中字符串的编码格式
为 gb2312
?>
```

注意：

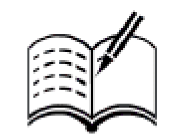
在连接数据库的过程中，要注意数据库服务器、用户名和密码的使用，还有连接的数据库要正确，以及对数据库中字符串编码格式的转换。

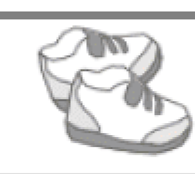
最后实现网站访问信息的存储，并且将访问量和 IP 访问量以图形图像的形式进行输出。

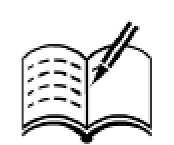
数据库数字计数器存储于 mr\4\04\文件夹下，其关键实现步骤如下。

（1）初始化一个 SESSION 变量，与数据库建立连接，获取当前时间和客户端的 IP 地址，并判断 SESSION 变量的值是否为空。

```
<?php
session_start();
include("conn/conn.php");
$data1=date("Y-m-d h:m:s");          //获取当前访问时间
$data2=date("Y-m-d");                //获取当前访问时间
$ip=getenv('REMOTE_ADDR');
if($_SESSION['temp']==""){           //判断$_SESSION[temp]==""的值是否为空，其中的
```







Note

temp 为自定义变量

（2）通过数据库存储网站被访问的数据，执行 SELECT 查询，判断当前获取的 IP 地址在数据库中是否已经存在。如果数据库中存在该 IP 地址，则执行 UPDATE 更新语句，将该 IP 地址对应的数据的访问量加 1，并且更新访问的时间；如果数据库中不存在该 IP 地址，则执行 INSERT 添加语句，将该 IP 地址的数据添加到数据库中。

```
$query=mysql_query("select * from tb_count04 where ip='$ip'");
$result=mysql_num_rows($query);
if($result>0){
$query_1=mysql_query("update  tb_count04  set  counts=counts+1,data1='$data1',data2=
'$data2' where ip='$ip'");
        }else{
            $query="insert into tb_count04(counts,data1,data2,ip)values ('1','$data1','$data2','$ip')";
          $result=mysql_query($query);
        }
```

（3）将 SESSION 变量赋值为 1，防止因页面刷新导致重复计数。

```
        $_SESSION['temp']=1;        //登录以后，$_SESSION[temp]的值不为空，给
$_SESSION[temp]赋一个值 1
}
?>
```

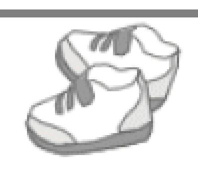
至此，数据库数字计数器介绍完毕。

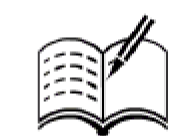
4.6　网站访问量统计分析

计数器中的有些功能已经在前面的章节中讲解过了，例如采用数据库来统计网站的访问量；通过 SESSION 来防止网站的重复计数；应用图形数字计数器来输出网站当前的访问量等。下面介绍如何对网站访问量的数据进行统计分析。这里主要应用 jpgraph 类创建柱形图，对网站访问量的统计结果进行分析，如图 4.8 所示。



图 4.8　对网站访问量的统计结果进行分析





4.6.1　通过数据库记录网站访问量

有关通过数据库统计网站访问量的具体方法已经在 4.5 节中作了详细讲解，这里不再赘述。输出数据库中存储的网站访问量应用的是 4.3 节的图形数字计数器，其关键代码如下：

```
<?php
    //以图形的形式输出数据库中的记录数
    $querys="select sum(counts) as ll from tb_count10 ";//查询数据库中总的访问量
    $results=mysql_query($querys,$id);
    $fwl=mysql_result($results,0,'ll');
    echo "----------";
    //对补位数字 0 的处理
    $len=strlen($fwl);                               //获取字符串的长度
    $str=str_repeat("0",6-$len);                     //获取 6-$len 个数字 0
    for($i=0;$i<strlen($str);$i++){                  //获取变量$str 的字符串长度
        $resultes=$str[$i];
        $resultes='<img src=images/0.gif>';
        echo $resultes;                              //循环输出$result 的结果
    }
    //对数据库中数据的处理
    for($i=0;$i<strlen($fwl);$i++){                  //获取字符串的长度
        $result=$fwl[$i];
        switch($result){
            //如果值为"0",则输出 0.gif 图片
        case "0"; $ret[$i]="0.gif";break;
            case "1"; $ret[$i]="1.gif";break;
            case "2"; $ret[$i]="2.gif";break;
            case "3"; $ret[$i]="3.gif";break;
            case "4"; $ret[$i]="4.gif";break;
            case "5"; $ret[$i]="5.gif";break;
            case "6"; $ret[$i]="6.gif";break;
            case "7"; $ret[$i]="7.gif";break;
            case "8"; $ret[$i]="8.gif";break;
            case "9"; $ret[$i]="9.gif";break;
        }
        echo "<img src=images/".$ret[$i].">";   //输出访问次数
    }
?>
```

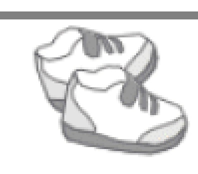
网站访问量的输出结果如图 4.9 所示。

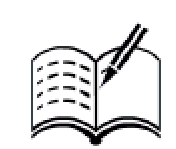


图 4.9　输出网站访问量

4.6.2　统计分析网站访问量

在图 4.9 所示的网站访问量输出结果中，有一个“网站访问量统计分析”超链接，它





链接到 indexs.php 文件中，在此文件中完成对网站访问量的统计操作，其统计的内容如图 4.10 所示。



图 4.10　网站访问量统计

具体实现步骤如下。

（1）连接数据库，调用指定的数据库连接文件。

```
<?php   include("conn/conn.php"); ?>
```

（2）输出网站总的访问量以及总的 IP 访问量。应用 SELECT 语句和 sum()函数，从数据表中读取 counts 字段总的数据，获取网站总的访问量。

```
<td width="240" height="25" align="center" valign="middle">
<?php
    $query_1="select sum(counts) as countes from tb_count10 ";
     //查询数据库中总的访问量
    $result_1=mysql_query($query_1);
    $countes_1=mysql_result($result_1,0,'countes');
     echo "<p class='STYLE1'>网站总的访问量：$countes_1</p>";
```

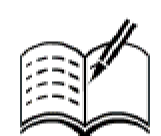
（3）获取网站总的 IP 访问量。从数据表中读取数据，应用 mysql_fetch_array()函数获取 ip 字段对应的 IP 地址，并将获取到的值赋给一个数组，通过 array_unique()函数去除数组中重复的值，最终获取到网站总的 IP 访问量。

```
     //查询数据库中总的 IP 访问量
    $query_2="select * from tb_count10 ";
    $result_2=mysql_query($query_2);
     while($myrow=mysql_fetch_array($result_2)){
          $counts_2[]=$myrow['ip'];                //将获取的 ip 值赋给变量
     }
     $results_2=array_unique($counts_2);           //去除数组中重复的值
     $countes_2=count($results_2);                 //获取数组中值的数量，即总的 IP 访问量
     echo "<p class='STYLE1'>网站总的 IP 访问量：$countes_2</p>";
?></td>
```

（4）获取网站当月的访问量和当月 IP 的访问量。使用的方法与获取网站总的访问量和总的 IP 访问量的方法类似，区别在于 SELECT 语句中增加了一个查询条件 data2='".substr(date("Y-m-d"),0,7)."'，即查询数据库中符合当前月份的数据。

```
    <td width="240" align="center" valign="middle">
<?php
    //查询数据库中当月总的访问量
```






```
        $query_3="select sum(counts) as countes from tb_count10 where data2='".substr
(date("Y-m-d"),0,7)."'";
        $result_3=mysql_query($query_3);
        $countes_3=mysql_result($result_3,0,'countes');
        echo "<p class='STYLE2'>网站当月的访问量：$countes_3</p>";
        //查询数据库中当月的 IP 访问量
        $query_4="select * from tb_count10 where data2='".substr(date("Y-m-d"),0,7)."'";
        $result_4=mysql_query($query_4);
        $counts_4=array();
        while($myrow_4=mysql_fetch_array($result_4)){
            $counts_4[]=$myrow_4['ip'];                    //将获取的 IP 值赋给变量
        }
        $results_4=array_unique($counts_4);                //去除数组中重复的值
        $countes_4=count($results_4);            //获取数组中值的数量，即总的 IP 访问量
        echo "<p class='STYLE2'>网站当月 IP 的访问量：$countes_4</p>";
?></td>
```

（5）获取网站当日的访问量和当日的 IP 访问量。其方法这里不再赘述，指定的查询条件是 data1='".date("Y-m-d")."'。程序关键代码如下：

```
        <td width="248" align="center" valign="middle">
<?php
        //查询数据库中当日总的访问量
        $query_5="select sum(counts) as countes from tb_count10 where data1='".date ("Y-m-d")."' ";
        $result_5=mysql_query($query_5);
        $countes_5=mysql_result($result_5,0,'countes');
        echo "<p class='STYLE3'>网站当日的访问量：$countes_5</p>";
        //查询数据库中当日的 IP 访问量
        $query_6="select * from tb_count10 where data1='".date("Y-m-d")."'";
        $result_6=mysql_query($query_6);
        $counts_6=array();
        while($myrow_6=mysql_fetch_array($result_6)){
            $counts_6[]=$myrow_6['ip'];                    //将获取的 IP 值赋给变量
        }
if(is_array($counts_6)){
        $results_6=array_unique($counts_6);                //去除数组中重复的值
        $countes_6=count($results_6);          //获取数组中值的数量，即总的 IP 访问量
        echo "<p class='STYLE3'>网站当日 IP 的访问量：$countes_6</p>";
}
<d>
```

4.6.3 柱形图分析网站月访问量比重

在 indexs.php 文件中，通过下拉列表框选择指定的月份，单击“提交”按钮，将在本页以柱形图的形式输出网站指定月份的访问量比重。

首先，创建表单，提交要分析的指定月份。其关键代码如下：

```
<form name="form1" method="post" action="indexs.php">
        <select name="select">
```




Note

```
    <?php
     $que="select distinct data2 from tb_count10";        //查询数据库中月份的数据
     $res=mysql_query($que,$id);
     while($myrow=mysql_fetch_array($res)){              //循环输出月份数据
     ?>
          <option value="<?php echo $myrow['data2'];?>"><?php echo $myrow['data2'];?></option>
          <?php
     }
          ?>
     </select>  
    <input type="submit" name="Submit" value="提交">
</form>
```

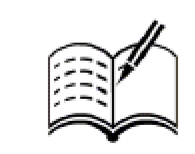
然后，在本页中获取表单提交的月份数据，以表单提交的数据为条件进行查询，将从数据库中读取的数据存储到指定的数组中，并将数组重新生成新字符串，将新字符串作为超链接的参数值传递到 stat.php 文件中。其关键代码如下：

```
<?php
if(isset($_POST['select'])){                                //判断提交的数据是否为真
$query="select counts,data1 from tb_count10 where data2='".$_POST['select']."' order by data1 ";                                  //定义 SQL 语句
     $result=mysql_query($query);                            //执行查询语句
     $results=array();                                       //定义数组
     $datas=array();                                         //定义数组
     while($myrow=mysql_fetch_array($result)){               //循环输出查询结果
          $results[]=$myrow['counts'];                       //将查询结果存储到数组中
          $lmbs=implode(",",$results);                       //将数组重新生成一个字符串
          $datas[]=substr($myrow['data1'],8,2);              //截取时间数据，并存储到数组中
          $das=implode(",",$datas);                          //将数组重新生成一个字符串
     }
?>
<table width="902" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#D5D5D5">
     <tr>
     <td height="18" background="images/mysql_9.jpg" bgcolor="#FFFFFF" class="STYLE4">
               <?php echo $_POST['select'];?>月网站访问量走势图
          </td>
     </tr>
    <tr>
     <td bgcolor="#FFFFFF">
          <p><img src="stat.php?lmbs=<?php echo $lmbs; ?>&das=<?php echo $das;?>"/></p>
          <p align="center" class="STYLE2"><?php echo $_POST['select'];?>月网站访问量走势图</p>
          </td>
     </tr>
</table>
```



```
<?php
}
?>
```

最后，在 stat.php 文件中，应用 jpgraph 类库创建柱形图，以超链接传递的参数值作为数据，生成网站月访问量统计图。stat.php 文件的代码如下：




```
<?php
include ("src/jpgraph.php");                                    //载入类库文件
include ("src/jpgraph_bar.php");
$lmbs=$_GET['lmbs'];                                             //获取超链接传递的参数值
$das=$_GET['das'];
$datay=explode(",",$lmbs);                     //按照","分隔字符串，$data 为获取的数据变量
$datax=explode(",",$das);                      //按照","分隔字符串，$data 为获取的数据变量
$graph = new Graph(800,200,"auto");                              //创建图像
$graph->img->SetMargin(60,20,30,50);                             //设置图像边框间距
$graph->SetScale("textlin");                                     //定义坐标刻度类型
$graph->SetMarginColor("lightblue");                             //定义图像颜色
$graph->title->Set('counter');                                   //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                      //设置标题字体
$graph->xaxis->SetFont(FF_VERDANA,FS_NORMAL,10);    //设置 X 轴的字体
$graph->yaxis->SetFont(FF_VERDANA,FS_NORMAL,10);    //设置 Y 轴的字体
$graph->xaxis->SetTickLabels($datax);                   //设置 X 轴输出的数据
$graph->xaxis->SetLabelAngle(50);                       //设置输出文字大小
$bplot = new BarPlot($datay);                           //实例化图像创建类
$bplot->SetWidth(0.6);                                  //设置柱形图的输出大小
$bplot->SetFillGradient("navy","#FFFF00",GRAD_LEFT_REFLECTION);       //设置图像的类型和填充颜色
$bplot->SetColor("white");                              //设置图像边框颜色
$graph->Add($bplot);                                    //添加数据
$graph->Stroke();                                       //生成图像
?>
```

4.7　技术提炼

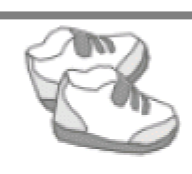
网页计数器的设计方法很多，应用不同方法设计的计数器的实效性也不尽相同。下面对网页计数器开发中应用到的技术进行提炼。

4.7.1　防止重复计数

在开发网页计数器的程序中，一个最为关键的问题就是如何防止网站的重复计数。如果一个计数器不能控制重复计数，那么这个计数器将没有任何意义。根据设计计数器的方法不同，总结出两种控制重复计数的方法。

1. 通过 SESSION 防止重复计数

此方法可以应用到任意一个计数器中实现控制重复计数的功能。其实现的原理是：在



当前页被访问时，初始化一个 SESSION 变量，并赋给其一个空值；然后判断 SESSION 变量的值是否为空，如果为空，则将计数器的值加 1，并且为 SESSION 变量赋值 1。

此时，在当前页中，SESSION 变量的值已经不为空，无论如何刷新，其值都不会改变，所以计数器的值也不会增加。关键代码如下：

```
<?php
//使用文本存储数据
if(!isset($_SESSION['temp'])){   //判断$_SESSION[temp]==""的值是否为空，其中的 temp
为自定义变量
    if(($fp=fopen("counter.txt","r"))==false){
        echo "打开文件失败!";
    }else{
        $counter=fgets($fp,1024);                    //读取文件中数据
        fclose($fp);                                 //关闭文本文件
        $counter++;                                  //计数器的值加 1
        $fp=fopen("counter.txt","w");                //以写的方式打开文本文件
        fputs($fp,$counter);                         //将新的统计数据加 1
        fclose($fp);
    }                                                //关闭文件
        $_SESSION['temp']=1; //计数器的值增加后，为$_SESSION[temp]赋值 1
}
//从文本文件中读取统计数据
if(($fp=fopen("counter.txt","r"))==false){
    echo "打开文件失败!";
}else{
    $counter=fgets($fp,1024);
    fclose($fp);
}
?>
```

注意：

SESSION 变量防止重复计数的缺点是只对当前页面有效，当重新打开该页面时，计数器的值仍然会增加。

2. 通过 IP 防止重复计数

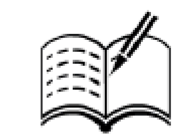
通过 IP 防止重复计数是最准确的方法，它完全是通过客户端机器的 IP 来防止重复计数，不会有任何时间和页面的限制，只要是已经访问过的 IP，计数器的值就不会增加。

通过 IP 防止重复计数必须要有数据库的支持，服务器会将每个访问过本网站的客户端的 IP 存储到数据库中。

当客户端的机器访问本网站时，服务器将首先获取该客户端的 IP 地址，并且与数据库中存储的 IP 地址进行比较，如果数据库中已经存在此 IP，则计数器的值将不会增加；否则将计数器的值加 1，并且将该 IP 地址存储到数据库中。关键代码如下：

```
<?php
session_start();
```






```
include("conn/conn.php");
$data1=date("Y-m-d h:m:s");            //获取当前访问时间
$data2=date("Y-m-d");                  //获取当前访问时间
$ip=getenv('REMOTE_ADDR');
//使用数据库存储数据
$query=mysql_query("select * from tb_count04 where ip='$ip'");
$result=mysql_num_rows($query);
if($result>0){
        echo "该 IP 地址已经存在";
}else{
        $query="insert into tb_count04(counts,data1,data2,ip)values('1','$data1','$data2','$ip')";
       $result=mysql_query($query);
}
?>
```

通过 IP 不但可以防止重复计数，而且可以统计出网站被同一 IP 访问的次数。

4.7.2　通过 array_unique()函数获取网站当月 IP 访问量

通过 array_unique()函数去除数组中重复的值，可以实现统计网站当月 IP 访问量。其中主要用到了 array_unique()函数和 count()函数。

（1）array_unique()函数

该函数将值作为字符串排序，然后对每个值只保留第一个键名，忽略后面所有的键名。其语法格式如下：

```
array array_unique ( array array)
```

（2）count()函数

该函数用于统计数组中元素值的个数。其语法格式如下：

```
int count ( mixed array [, int mode])
```

参数说明如下。

☑　array：指定的数组。

☑　mode：可选参数，COUNT_RECURSIVE（或 1）。如选中此参数，本函数将递归地对数组计数。这对于计算多维数组的所有单元尤其有用。此参数的默认值是 0。

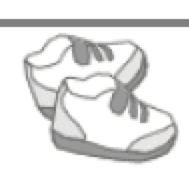
在获取网站当月 IP 访问量之前，先来看一下存储网站访问量数据的数据表的结构，如图 4.11 所示。

服务器: localhost ▸ 数据库: db_counter ▸ 表 : tb_count10

浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

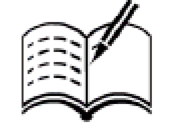
	字段	类型	整理	属性	Null	默认	额外	操作
☐	tb_id	int(10)			否		auto_increment	
☐	counts	varchar(50)	gb2312_chinese_ci		否			
☐	data1	varchar(50)	gb2312_chinese_ci		否			
☐	data2	varchar(50)	gb2312_chinese_ci		否			
☐	ip	varchar(50)	gb2312_chinese_ci		否			

图 4.11　网站访问量统计数据表



在该数据表中，包括 5 个字段，分别是 ID；网站访问量 counts；访问时间 data1，完整的时间年月日；访问时间 data2，只包括年和月；网站的 IP 地址。

要获取网站当月 IP 访问量，其查询的条件必须从时间 data2 着手，查询出当前时间符合 data2 的数据。


Note

```
<?php
$query_4="select * from tb_count10 where data2='".substr(date("Y-m-d"),0,7)."'";
    $result_4=mysql_query($query_4);
```

技巧：

应用 substr()函数，获取当前日期中年和月的值，将获取的值与数据表中 data2 字段的值进行比较，从中读取出符合条件的数据。

然后，将获取到的 IP 地址写入到一个数组中。

```
while($myrow_4=mysql_fetch_array($result_4)){
    $counts_2[]=$myrow['ip'];                    //将获取的 ip 值赋给变量
}
```

最后，应用 array_unique()函数去除数组中重复的值，再统计出数组中元素的数量，即总的 IP 访问量。

```
$results_4=array_unique($counts_4);          //去除数组中重复的值
$countes_4=count($results_4);                //获取数组中值的数量，即总的 IP 访问量
?>
```

4.7.3　将数组中的数据写入到字符串中

将数组中的数据写入到字符串中，主要应用的是 current ()函数和 implode ()函数。

（1）current()函数

每个数组中都有一个内部指针，初始指向插入到数组中的第一个单元。该函数返回当前被内部指针指向的数组单元的值，并不移动指针。如果内部指针指向超出了数组的最大单元，将返回 false。其语法格式如下：

```
mixed current ( array array)
```

（2）implode()函数

该函数用于将数组的内容组合成一个字符串。其语法格式如下：

```
string implode(string glue, array pieces)
```

参数 glue 是字符之间的分隔符。

这里实际是将指定月份每天的数据写入到字符串中，实现对网站指定月份的访问量进行分析。要完成该操作，首先要以指定的月份为查询条件，查询出指定月份的网站访问量的数据。

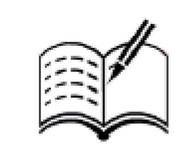
```
<?php
```



```
$query="select counts,data1 from tb_count10 where data2='".$_POST['select']."' order by data1 ";
$result=mysql_query($query);
```

然后，通过 while 语句，循环读取其中的数据，并且应用 current()函数，将数组中的当前元素写入到数组$results 中。

最后应用 implode()函数，将数组$results 中的数据写入到一个字符串变量$lmbs 中，以“,”为分隔符，完成将指定月份每天的数据写入到字符串中的操作。



```
    while($myrow=mysql_fetch_array($result)){
        $results[]=$myrow['counts'];
        $lmbs=implode(",",$results);
    }
?>
```

4.7.4　动态生成柱形图分析网站访问量

动态生成柱形图分析网站访问量，主要是应用 Jpgraph 类库，根据从数据库中读取的数据生成柱形图，完成对网站访问量的分析。其前提是必须在项目中载入 Jpgraph 类库。由于其属于第三方组件，在配书光盘的源代码中没有提供，所以需要读者自行下载，并且将下载的 src 文件夹复制到项目 mr\4\05 文件夹下，然后才可以完成动态生成柱形图分析网站访问量的操作。

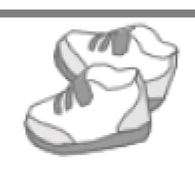
第 5 章

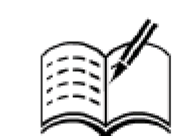
购物车模块

（自学视频、源程序：配套资源\mr\5\）

在网上购物时，经常会用到购物车。这是一种用于临时存储商品的工具，非常方便、实用。有了它，再也不用担心一次购买多个商品时要进行多次提交结算的操作，可以将其逐一放入购物车中，等选购完所有商品之后，再一起进行结算。购物车是电子商务类网站中一项必不可少的功能，其实现的方法也很多，这里将向广大读者介绍一种最为常用的购物车——SESSION 购物车。本章将详细地讲解 SESSION 购物车的创建和使用方法。通过本章的学习，读者能够学到以下内容。

- SESSION 购物车的创建
- 更改购物车中的商品数量
- 删除购物车中的商品
- 统计购物车中的商品金额
- 向购物车中添加商品
- 清空购物车
- 通过 PDO 操作 MySQL 数据库
- Smarty 模板实现网页动静分离





5.1　购物车模块概述

5.1.1　功能概述

这里的购物车模块是以电子商务网站为大背景进行开发的，不仅实现了购物车本身应该具备的功能，如添加商品、删除商品、更改商品数量、商品金额小计、商品金额总计和清空购物车等，而且还包括了电子商务网站的其他功能，如用户注册、用户登录、订单查询和商品分类等。其功能结构如图 5.1 所示。

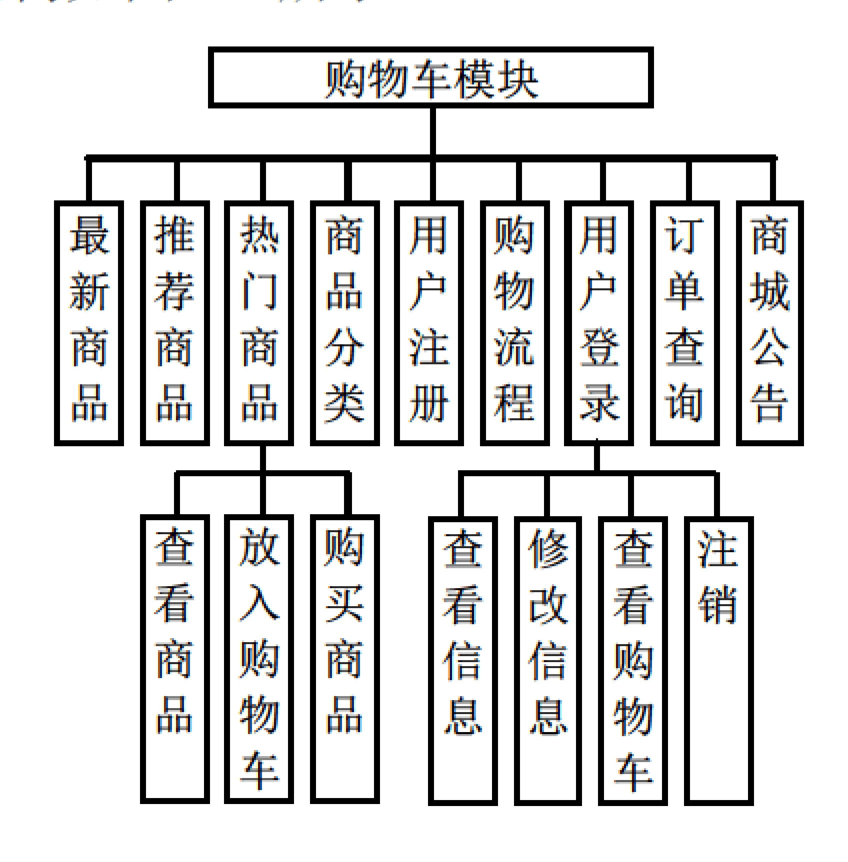


图 5.1　购物车模块功能结构图

5.1.2　购物车操作流程

购物车的操作流程：首先，登录到网站中浏览商品；然后，购买指定的商品，进入购物车页面中，在该页面可以实现很多操作，包括更改商品数量、删除商品、清空购物车、继续购物等；最后，依次执行填写收货人信息、生成订单、订单打印，以及提交订单等操作。其操作流程如图 5.2 所示。

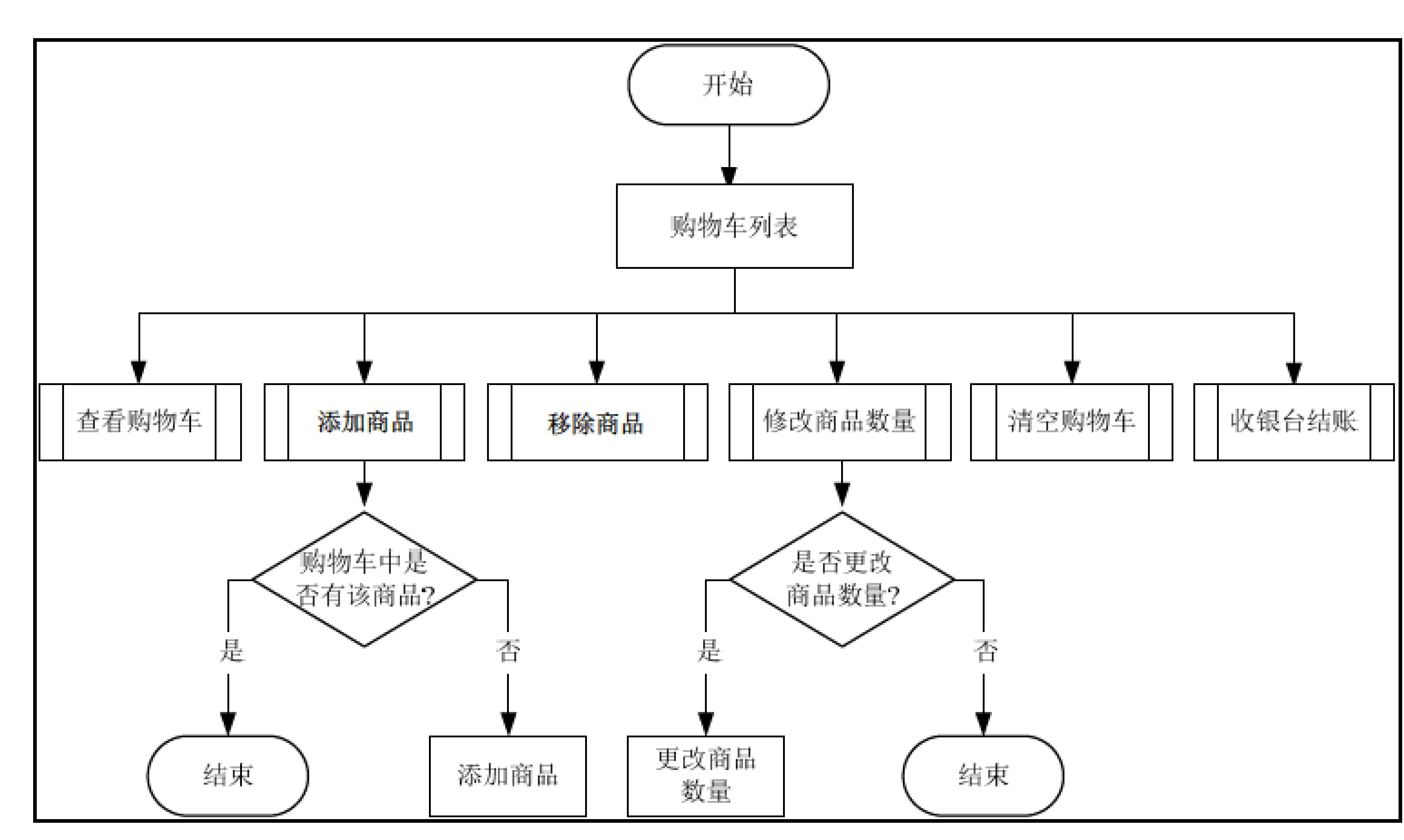


图 5.2　购物车操作流程



Note

5.1.3 程序预览

购物车模块只是电子商务网站中的一个子功能，为了让读者对本模块有个初步的了解和认识，下面列出几个典型功能的页面。

明日购物商城网站主页如图 5.3 所示，其中包括部分最新商品、热门商品、推荐商品以及网站的最新公告和会员登录窗口。



图 5.3 明日购物商城网站主页

推荐商品展示页面如图 5.4 所示，以分页的形式展示了网站的所有推荐商品。

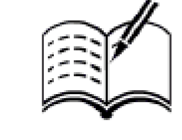


图 5.4 推荐商品展示页面

购物车页面如图 5.5 所示，展示了会员在本网站购买的商品，并提供了购物车的各种功能。







我的购物车						
	商品名称	购买数量	市场价格	会员价格	折扣率	合计
□	自行车	3	388	349.2	9	1047.6
□	数码相机	5	1888	1699.2	9	8496
□	洗衣机	1	2666	2399.4	9	2399.4
□	家庭影院	1	4888	4399.2	9	4399.2
全选 反选	删除选择		继续购物	去收银台		共计：16342.2 元

图 5.5　购物车页面

5.2　数据库设计

无论是什么系统软件，其最根本的功能就是对数据的操作与使用。因此，一定要先做好数据库的分析、设计与实现，然后才能实现对应的功能模块。

5.2.1　创建数据库

首先来看一下电子商务平台所使用的数据库与数据表的情况，如图 5.6 所示。


服务器: localhost ▸ 数据库: db_business

结构　SQL　搜索　查询　导出　Import　操作　权限　删除

表	操作	记录数	类型	整理	说明
tb_admin		2	MyISAM	utf8_unicode_ci	管理员表
tb_class		13	MyISAM	utf8_unicode_ci	商品类别表
tb_commo		4	MyISAM	utf8_unicode_ci	商品信息表
tb_form		17	MyISAM	utf8_unicode_ci	商品订单表
tb_links		2	MyISAM	utf8_unicode_ci	友情链接表
tb_opinion		0	MyISAM	utf8_unicode_ci	商品留言表
tb_public		4	MyISAM	utf8_unicode_ci	商品公告表
tb_user		6	MyISAM	utf8_unicode_ci	会员信息表
8 个表	总计	48	MyISAM	utf8_general_ci	

图 5.6　电子商务数据库与数据表

5.2.2　创建数据表

下面来看各个数据表的结构和字段说明。

☑　tb_user（会员信息表）

会员信息表主要用于存储注册会员信息，其结构如图 5.7 所示。

服务器: localhost ▸ 数据库: db_business ▸ 表 : tb_user

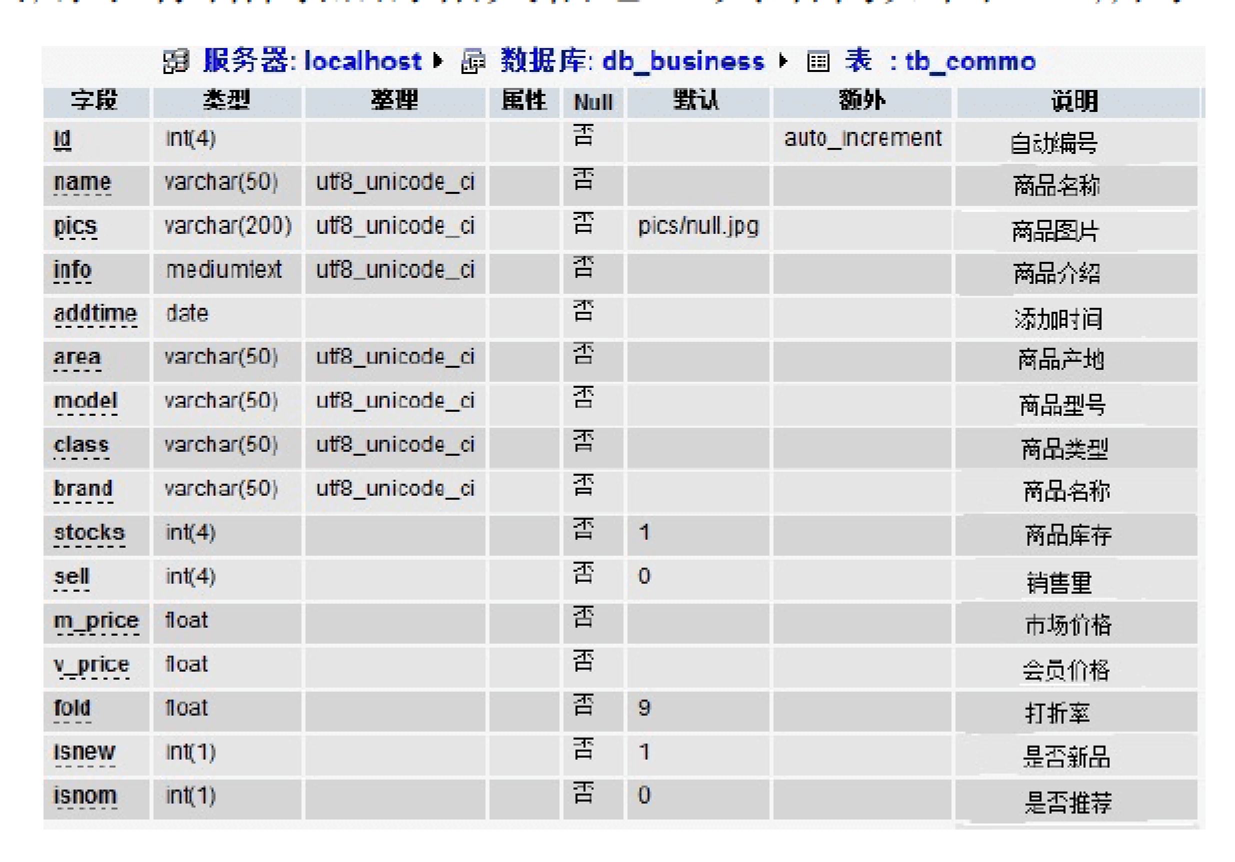
字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(80)	utf8_unicode_ci		否			用户名称
password	varchar(80)	utf8_unicode_ci		否			用户密码
question	varchar(80)	utf8_unicode_ci		否			密保问题
answer	varchar(80)	utf8_unicode_ci		否			密保答案
realname	varchar(80)	utf8_unicode_ci		否			用户真实姓名
card	varchar(80)	utf8_unicode_ci		否			身份证号
tel	varchar(80)	utf8_unicode_ci		否			手机号码
phone	varchar(80)	utf8_unicode_ci		否			座机号码
Email	varchar(80)	utf8_unicode_ci		否			用户邮箱
QQ	varchar(80)	utf8_unicode_ci		否			QQ号
code	varchar(80)	utf8_unicode_ci		否			邮编号码
address	varchar(200)	utf8_unicode_ci		否			地址
addtime	datetime			否			注册时间
isfreeze	int(1)			否	0		是否冻结
shopping	varchar(200)	utf8_unicode_ci		是	NULL		购物车信息
consume	float			否			

图 5.7　会员信息表结构



☑ tb_commo（商品信息表）

商品信息表主要用于存储商品的相关信息，其结构如图 5.8 所示。



服务器: localhost ▸ 数据库: db_business ▸ 表 : tb_commo

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	utf8_unicode_ci		否			商品名称
pics	varchar(200)	utf8_unicode_ci		否	pics/null.jpg		商品图片
info	mediumtext	utf8_unicode_ci		否			商品介绍
addtime	date			否			添加时间
area	varchar(50)	utf8_unicode_ci		否			商品产地
model	varchar(50)	utf8_unicode_ci		否			商品型号
class	varchar(50)	utf8_unicode_ci		否			商品类型
brand	varchar(50)	utf8_unicode_ci		否			商品名称
stocks	int(4)			否	1		商品库存
sell	int(4)			否	0		销售里
m_price	float			否			市场价格
v_price	float			否			会员价格
fold	float			否	9		打折率
isnew	int(1)			否	1		是否新品
isnom	int(1)			否	0		是否推荐

图 5.8　商品信息表结构

此外，还有管理员表、商品类别表、商品订单表、商品公告表、用户信息表、友情链接表和商品留言表等，限于篇幅，这里不再介绍，具体参见配套资源中的数据库文件。

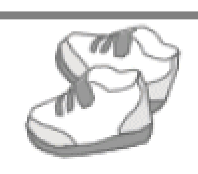
5.3　首 页 设 计

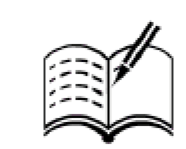
5.3.1　首页概述

首页一般没有多少实质的技术，主要是加载一些功能模块，如登录模块、导航栏模块、公告栏模块等，使浏览者能够了解网站内容和特点。首页设计的关键是要合理地对页面进行布局，既要尽可能地将重点模块显示出来，同时又不能因为页面凌乱无序，而让浏览者无所适从、产生反感。本模块首页的设计效果如图 5.9 所示。



图 5.9　首页设计效果



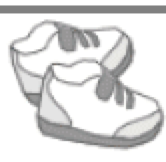


Note

5.3.2　首页实现过程

（1）创建 index.php 动态页。在 index.php 动态页中，应用 include_once()语句包含相应的文件；应用 switch 语句，以超链接中参数 page 传递的值为条件进行判断，实现在不同页面之间跳转。index.php 的关键代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
require("system/system.inc.php");                             //包含配置文件
if(isset($_GET["page"])){
        $page=$_GET["page"];
}else{
        $page="";
}
include_once("login.php");
include_once("public.php");
include_once("links.php");
switch($page){
        case "hyzx":
                include_once "member.php";
                $smarty->assign('admin_phtml','member.tpl'); //将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
        case 'allpub':
                include_once 'allpub.php';
                $smarty->assign('admin_phtml','allpub.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
        case 'nom':
                include_once 'allnom.php';
                $smarty->assign('admin_phtml','allnom.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
        case 'new':
                include_once 'allnew.php';
                $smarty->assign('admin_phtml','allnew.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
        case 'hot':
                include_once 'allhot.php';
                $smarty->assign('admin_phtml','allhot.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
        case 'shopcar':
                include_once 'myshopcar.php';
                $smarty->assign('admin_phtml','myshopcar.tpl');//将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
```




```
        case 'settle':
            include_once 'settle.php';
            $smarty->assign('admin_phtml','settle.tpl');    //将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
        case 'queryform':
            include_once 'queryform.php';
            $smarty->assign('admin_phtml','queryform.tpl');//将 PHP 脚本文件对应的模板文件名称赋给模板变量
        break;
        default:
            include_once 'newhot.php';
            $smarty->assign('admin_phtml','newhot.tpl');  //将 PHP 脚本文件对应的模板文件名称赋给模板变
        break;
    }
    $smarty->display("index.tpl");                         //指定模板页
    ?>
```

（2）创建 system\templates\index.tpl 模板页。在模板文件 index.tpl 中应用 Smarty 的 include 标签调用不同的模板文件，生成静态页面。其关键代码如下：

```
<table width="850" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td colspan="2">{include file='top.tpl'}</td>
  </tr>
  <tr>
    <td width="216" align="left" valign="top">
     {include file='login.tpl'}
     {include file='public.tpl'}
     {include file='links.tpl'}
    </td>
    <td width="634" height="700" align="center" valign="top">
{include file='search.tpl'}
<!--载入模板文件-->{include file=$admin_phtml}</td>
  </tr>
</table>
<table width="850" border="0" cellspacing="0" cellpadding="0">
     <tr>
     <td>{include file='buttom.tpl'}</td>
    </tr>
</table>
```

说明：

本模块的功能较多，结构比较复杂，对于初学者来说学起来可能会比较困难。所以，本书将系统中的各个功能模块所涉及的文件（如 PHP、TPL、CSS、JS 等）尽可能都单独实现。读者在学习其中某个模块时，可以将相关的文件统一放到同一个目录下单独测试。





5.4　登录模块设计

5.4.1　登录模块概述

登录模块是会员与系统进行交互的窗口。匿名用户虽然也可以访问本网站，但只能进行浏览、查询等简单操作，而会员则可以购买商品，并且能享受超低价格。登录模块包括用户注册和用户登录两部分，其运行效果如图 5.10 所示。



图 5.10　登录模块运行效果

5.4.2　用户注册

用户注册的主要功能是注册新会员。如果信息输入完整而且符合要求，则系统会将该用户信息保存到数据库中，否则显示错误原因，以便用户改正。用户注册页面的运行效果如图 5.11 所示。

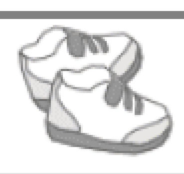


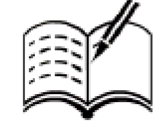
图 5.11　用户注册页面

具体实现过程如下。

（1）创建 register.tpl 模板文件，编写用户注册页面。其中包含两个 JS 脚本文件，即 createxmlhttp.js 和 check.js。其中，createxmlhttp.js 是 Ajax 的实例化文件，而 check.js 是对用户注册信息进行验证，并且返回验证结果。

（2）创建 register.php 动态 PHP 文件，加载模板。register.php 文件的代码如下：





Note

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );            //设置文件编码格式
require("system/system.inc.php");                               //包含配置文件
$smarty->assign('title','新用户注册');
$smarty->display('register.tpl');
?>
```

（3）创建 reg_chk.php 文件，获取表单中提交的数据，将数据存储到指定的数据表中。reg_chk.php 的代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );            //设置文件编码格式
require("system/system.inc.php");                               //包含配置文件
        $name = $_POST['name'];
        $password = md5($_POST['pwd1']);
        $question = $_POST['question'];
        $answer = $_POST['answer'];
        $realname = $_POST['realname'];
        $card = $_POST['card'];
        $tel = $_POST['tel'];
        $phone = $_POST['phone'];
        $Email = $_POST['email'];
        $QQ = $_POST['qq'];
        $code = $_POST['code'];
        $address = $_POST['address'];
        $addtime = date("Y-m-d H:i:s");
        $sql="insertinto
tb_user(name,password,question,answer,realname,card,tel,phone,Email,QQ,code,address,
addtime,isfreeze,shopping,consume)" ;
        $sql .= " values ('$name', '$password', '$question', '$answer', '$realname', '$card', '$tel',
'$phone', '$Email',
'$QQ', '$code', '$address','$addtime','0','','00.00')";
        $rst= $admindb->ExecSQL($sql,$conn);
        if($rst){
                $_SESSION['member'] = $name;
                echo "<script>top.opener.location.reload();alert('注册成功');window.close();
</script>";
        }else{
                echo '<script>alert(\'添加失败\');history.back;</script>';
        }
?>
```

（4）在 system\templates\login.tpl 模板页中，创建“用户注册”超链接。当用户单击前台的 注册 按钮时，系统会调用 JS 的 onclick 事件，弹出注册窗口。其代码如下：

```
<a href="#" id="login" onclick="reg()"><img src="images/check.JPG" width="59" height="23"
border="0" /></a>
```

这里用到的 JS 文件为 js/login.js，调用的函数为 reg()。该函数的代码如下：



```
function reg(){
	window.open("register.php", "_blank", "width=600,height=650",false);		//弹出窗口
}
```

5.4.3　用户登录

用户在输入正确的用户名、密码和验证码之后，将返回如图 5.12 所示的信息。



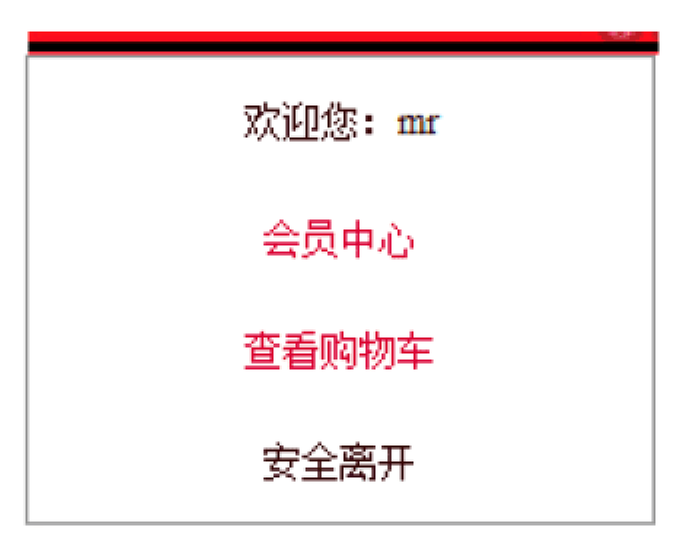
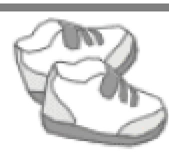


图 5.12　用户成功登录效果

具体实现过程如下。

（1）创建模板文件 login.tpl，完成用户登录表单的设计。在该页面中，当单击 Submit 按钮时，系统将调用 lg()函数对用户提交信息进行验证。lg()函数包含在 js/login.js 脚本文件内，其代码如下：

```
// JavaScript Document
function lg(form){
	if(form.name.value==""){
		alert('请输入用户名');
		form.name.focus();
		return false;
	}
	if(form.password.value == "" || form.password.value.length < 6){
		alert('请输入正确密码');
		form.password.focus();
		return false;
	}
	if(form.check.value == ""){
		alert('请输入验证码');
		form.check.focus();
		return false;
	}
	if(form.check.value != form.check2.value){
		form.check.select();
		code(form);
		return false;
	}
	var user = form.name.value;
	var password = form.password.value;
	var url = "chkname.php?user="+user+"&password="+password;
	xmlhttp.open("GET",url,true);
	xmlhttp.onreadystatechange = function(){
```

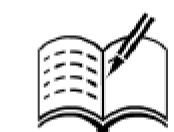



```
        if(xmlhttp.readyState == 4){
                var msg = xmlhttp.responseText;
                if(msg == '1'){
                    alert('用户名或密码错误!!');
                    form.password.select();
                    form.check.value = '';
                    code(form);
                    return false;
                }if(msg == "3"){
                    alert("该用户被冻结，请联系管理员");
                    return false;
                }else{
                    alert('欢迎光临');
                    location.reload();
                }
            }
        }
        xmlhttp.send(null);
        return false;
    }
    //显示验证码
    function yzm(form){
        var num1=Math.round(Math.random()*10000000);
        var num=num1.toString().substr(0,4);
        document.write("<img name=codeimg width=65 heigh=35 src='yzm.php?num="+num+"'>");
        form.check2.value=num;
    }
    //刷新验证码
    function code(form){
        var num1=Math.round(Math.random()*10000000);
        var num=num1.toString().substr(0,4);
        document.codeimg.src="yzm.php?num="+num;
        form.check2.value=num;
    }
    //注册
    function reg(){
    window.open("register.php", "_blank", "width=600,height=650",false);
    }
    //找回密码
    function found() {
    window.open("found.php","_blank","width=350 height=240",false);
    }
```

用户名和密码是在 chkname.php 页面中被验证。

（2）创建用户信息模板文件 system\templates\info.tpl。用户登录成功后，在原登录框位置将显示用户信息。用户可以通过“会员中心”超链接对自己的信息进行修改，也可以单击“查看购物车”超链接查看购物车商品；当用户离开时，可以单击“安全离开”超链





接。用户信息模块的主要代码如下：

```
<!--显示当前登录用户名-->
欢迎您：{$member}
<!--会员中心超链接-->
<a href="?page=hyzx" id="info" class="lk">会员中心</a>
<!--查看购物车-->
<a href="?page=shopcar" class="lk">查看购物车</a>
<!--安全离开-->
<a onclick="javascript:logout()" style="cursor:hand" id="info">安全离开</a>
```

5.5　商品展示模块设计

5.5.1　商品展示模块概述

本系统为用户提供了几种不同的商品展示方式，包括推荐商品、最新商品、热门商品等，能够使消费者有目的地选购商品。每种展示方式中均包括商品的详细信息显示，为用户购买商品提供可靠的依据。商品展示模块的运行效果如图 5.13 所示。



图 5.13　商品展示页面

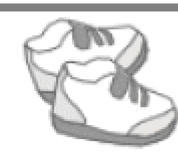
因为“推荐商品”、“最新商品”和“热门商品”模块的实现方法和过程基本相同，在此以“推荐商品”模块为例进行介绍，其他模块相关代码可参见配套资源中的源程序。

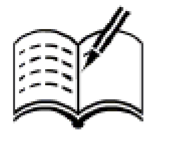
5.5.2　商品展示模块

在技术分析中已经对商品展示所使用的技术、方法作了概述，下面再来介绍一下其具体的实现过程。

（1）创建 allnom.php 文件，从数据库中读取推荐商品的数据，并将其存储到模板变量中。具体代码参见 5.7.6 节。

（2）创建 allnom.tpl 模板页，应用 section 语句输出商品信息，并添加相应的操作按钮或超链接。模板页中一共有两个事件：查看商品详细信息和放入购物车。





Note

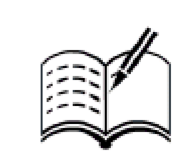
☑ 当单击“查看详情”按钮时，将触发 onclick 事件，并将调用 openshowcommo()函数，同时商品 id 会作为函数的唯一参数被传递进去。

☑ 当单击“购买”按钮时，同样会触发 onclick 事件，并调用 buycommo()函数，唯一的参数也是商品的 id。

商品模板页面 system\templates\allnom.tpl 的代码如下：

```
<table width="636" border="0" align="center" cellspacing="0" cellpadding="0">
  <tr>
    <td  height="33"  align="left"  valign="middle"  background="images/shop_07.gif"> </td>
  </tr>
  <tr>
    <td height="132" align="left" valign="middle">
   {section name=nom_id loop=$nomarr}
    <table width="636" height="134" border="0" cellspacing="0" cellpadding="0">
     <tr>
          <td width="145" rowspan="5" align="center" valign="middle">
<img src="{$nomarr[nom_id].pics}" alt="{$nomarr[nom_id].name}" style="border: 1px solid #f0f0f0;" /></td>
              <td height="35">商品名称：{$nomarr[nom_id].name}</td>
              <td width="156" height="35">商品类别：{$nomarr[nom_id].class}</td>
         <td width="157">商品型号：{$nomarr[nom_id].model}</td>
     </tr>
     <tr>
       <td height="23">商品品牌：{$nomarr[nom_id].brand}</td>
            <td height="23" colspan="2">商品产地：{$nomarr[nom_id].area}</td>
     </tr>
     <tr>
       <td width="178" height="23">剩余数量：{$nomarr[nom_id].stocks}</td>
            <td colspan="2">销售数量：{$nomarr[nom_id].sell}</td>
     </tr>
     <tr>
       <td  height="23">市场价：<font  color="red">{$nomarr[nom_id].m_price} 元</font></td>
            <td height="23" colspan="2">上市日期：{$nomarr[nom_id].addtime}</td>
     </tr>
     <tr>
       <td  height="30">会员价格：<font  color="#FF0000">{$nomarr[nom_id].v_price} 元</font></td>
            <td height="30" colspan="2" align="center" valign="middle">
<input id="allshow" name="allshow" type="button" value=""
class="showinfo" onclick="openshowcommo({$nomarr[nom_id].id})"/>;
<input id="buy" name="buy" type="button" value="" class="buy"
onclick="return buycommo({$nomarr[nom_id].id})" /></td>
     </tr>
    </table>
    <hr style="border: 1px solid #f0f0f0;" />
    {/section}
    </td>
```






```
    </tr>
    <tr>
        <td height="25">{$rst1_page}</td>
    </tr>
</table>
```

（3）创建 showcommo.js 脚本文件。当单击“查看详情”按钮时，系统会弹出一个新的页面，并显示商品的详细信息；当单击“购买”按钮时，该商品将会被放到当前用户的购物车中，如果没有登录或商品已添加，则会提示错误信息。showcommo.js 脚本文件的代码如下：

```
/*查看商品信息函数，将打开一个新页面*/
function openshowcommo(key){
    open('showcommo.php?id='+key,'_blank','width=560 height=300',false);
}
/*将购买商品添加到购物车中，将在下节中讲解*/
function buycommo(key){
    …
}
```

5.6　购物车模块设计

5.6.1　购物车模块概述

购物车是电子商务平台中非常关键的一个功能模块，其主要功能是保留用户选择的商品信息，用户可以在购物车内设置选购商品的数量，显示选购商品的总金额，还可以清除选择的全部商品信息，重新选择商品。购物车页面运行效果如图 5.14 所示。

我的购物车						
	商品名称	购买数量	市场价格	会员价格	折扣率	合计
☑	液晶显示器	1	5888	5299.2	9	5299.2
☑	自行车	3	388	349.2	9	1047.6
☑	家庭影院	2	4888	4399.2	9	8798.4
全选 反选	删除选择		继续购物	去收银台		共计：15144 元

图 5.14　购物车页面

购物车模块除了展示用户购买的商品之外，还可实现添加商品、更改商品数量、删除商品和清空购物车等操作。

5.6.2　购物车展示

购物车页面分为 PHP 代码页和 Smarty 模板页。在 PHP 代码页中，首先读取 tb_user 数据表中 shopping 字段的内容，如果字段为空，则输出“暂无商品”；如果数据库中有数据，则循环输出数据，并将商品信息保存到数组中，再传给模板页。购物车页面 myshopcar.php 的代码如下：

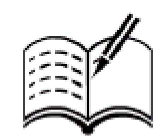
Note

```
<?php
$select = "select id,shopping from tb_user where name ='".$_SESSION['member']."'";
$rst = $admindb->ExecSQL($select,$conn);                    //执行查询语句
if($rst[0]['shopping']==""){                                 //判断查询结果
    echo "<script>alert('购物车中暂时没有商品!');window.location.href='index.php';
</script>";
}
$commarr = array();
foreach($rst[0] as $value){                                  //循环读取内容
    $tmpnum = explode('@',$value);
    $shopnum = count($tmpnum);                               //商品类数
    $sum = 0;
    foreach($tmpnum as $key => $vl){
        $s_commo = explode(',',$vl);
        $sql2 = "select id,name,m_price,fold,v_price from tb_commo";
        $commsql = $sql2." where id = ".$s_commo[0];      //定义查询语句
        $arr = $admindb->ExecSQL($commsql,$conn);        //执行查询语句
        @$arr[0]['num'] = $s_commo[1];
        @$arr[0]['total'] = $s_commo[1]*$arr[0]['v_price'];
        $sum += $arr[0]['total'];
        $commarr[$key] = $arr[0];                           //统计结果数据
    }
}
$smarty->assign('shoparr',$shopnum);                         //将数据赋给指定的模板变量
$smarty->assign('commarr',$commarr);
$smarty->assign('sum',$sum);
?>
```

购物车模板页 system\templates\myshopcar.tpl 不仅要负责用户购买商品信息的输出，而且还要提供可以对商品进行修改、删除等操作的事件接口。模板页代码如下：

```
<table border="0" cellspacing="0" cellpadding="0" align="center">
<form id="myshopcar" name="myshopcar" method="post" action="#">
  <tr>
    <td height="30" colspan="7" align="center" valign="middle" class="first">我的购物车</td>
  </tr>
  <tr>
    <td width="35" height="25" align="center" valign="middle" class="left"> </td>
    <td width="100" height="25" align="center" valign="middle" class="center">商品名称</td>
    <td width="100" height="25" align="center" valign="middle" class="center">购买数量</td>
    <td width="100" height="25" align="center" valign="middle" class="center">市场价格</td>
    <td width="100" height="25" align="center" valign="middle" class="center">会员价格</td>
    <td width="100" height="25" align="center" valign="middle" class="center">折扣率</td>
    <td width="100" height="25" align="center" valign="middle" class="right">合计</td>
  </tr>
{foreach key=key item=item from=$commarr}
  <tr>
    <td height="25" align="center" valign="middle" class="left">
<input id="chk" name="chk[]" type="checkbox" value="{$item.id}"></td>
```





```
        <td height="25" align="center" valign="middle" class="center">
<div id = "c_name{$key}">  {$item.name}</div></td>
        <td height="25" align="center" valign="middle" class="center">
<input id="cnum{$key}" name="cnum{$key}" type="text" class="shorttxt"
value="{$item.num}"
onkeyup="cvp({$key},{$item.v_price},{$shoparr})"></td>
        <td height="25" align="center" valign="middle" class="center">
<div id="m_price{$key}"> {$item.m_price}</div></td>
        <td height="25" align="center" valign="middle" class="center">
<div id="v_price{$key}"> {$item.v_price}</div></td>
        <td height="25" align="center" valign="middle" class="center">
<div id="fold{$key}"> {$item.fold}</div></td>
        <td height="25" align="center" valign="middle" class="right">
<div id="total{$key}"> {$item.total}</div></td>
    </tr>
{/foreach}
    <tr>
        <td height="25" colspan="3" align="left" valign="middle">
        <a href="#" onclick="return alldel(myshopcar)">全选</a>
<a href="#" onclick="return overdel(myshopcar);">反选</a>  
            <input type="button" value="删除选择" class="btn" style="border-color: #FFFFFF;"
onClick = 'return del(myshopcar);'>
      </td>
        <td height="25" align="center" valign="middle">
<input id="cont" name="cont" type="button" class="btn" value="继续购物"
onclick="return conshop(myshopcar)" /></td>
        <td height="25" align="center" valign="middle">
<input id="uid" name="uid" type="hidden" value="{$smarty.session.member}" >
<input id="settle" name="settle" type="button" class="btn" value="去收银台" onclick="return
formset(form)" /></td>
        <td height="25" colspan="2" align="right" valign="middle"><div id='sum'>共计：
{$sum} 元</div></td>
        </tr>
</form>
</table>
```

5.6.3　更改商品数量

对于新添加的商品，默认的购买数量为 1，在购物车页面中也可以对商品的数量进行修改。当商品数量发生变化时，商品的“合计”金额和商品总金额会自动发生改变，该功能是通过触发 text 文本域的 onkeyup 事件调用 cvp()函数实现的。cvp()函数有 3 个参数，分别是商品 id、商品单价和商品类别。

在 shopcar.js 文件中，首先通过商品的 id 得到要修改商品的相关表单和标签属性；然后通过商品单价和输入的商品数量计算该商品的合计金额；接着使用 for 循环得到其他商品的合计金额；最后将所有的合计金额累加，并输出到购物车页面。cvp()函数代码如下：

```
function cvp(key,vpr,shoparr){                    //更新商品数量
```




Note

```
        var n_pre = 'total';                              //定义变量
        var num = 'cnum'+key.toString();                  //获取传递的参数值
        var total = n_pre+key.toString();
        var t_number = document.getElementById(num).value;    //获取 ID 值
        var ttl = t_number * vpr;
        document.getElementById(total).innerHTML = ttl;
        var sm = 0;
        for(var i = 0; i < shoparr; i++){                     //循环输出
              var aaa = document.getElementById(n_pre+i.toString()).innerText;    //计算数据
              sm += parseInt(aaa);
        }
        document.getElementById('sum').innerHTML = '共计：'+sm+' 元';        //输出结果
   }
```

这里所更改的商品数量，并没有被保存到数据库中，如果希望保存，那么单击“继续购物”按钮，即可将商品数量更新到数据库中。

5.6.4　删除商品

当对添加的商品不满意时，可以对商品进行删除操作。操作流程为：首先选中要删除商品前面的复选框，如果全部删除，则可以单击“全选”按钮；然后单击“删除选择”按钮，在弹出的警告对话框中单击“确定”按钮，所选商品将被全部删除。删除商品页面的运行效果如图 5.15 所示。



图 5.15　删除商品流程

所有的删除操作都是通过 JS 脚本文件 shopcar.js 来实现，相关的函数包括 alldel()、overdel()和 del()。alldel()函数和 overdel()函数实现的原理比较简单，通过触发 onClick()事件来改变复选框的选中状态。函数代码如下：

```
//全部选择/取消
function alldel(form){
        var leng = form.chk.length;
        if(leng==undefined){
            if(!form.chk.checked)
                    form.chk.checked=true;
          }else{
            for( var i = 0; i < leng; i++)
```




```
                {
                        if(!form.chk[i].checked)
                                form.chk[i].checked = true;
                }
            }
        return false;
}
//反选
function overdel(form){
        var leng = form.chk.length;
        if(leng==undefined){
            if(!form.chk.checked)
                    form.chk.checked=true;
                else
                    form.chk.checked=false;
        }else{
            for( var i = 0; i < leng; i++)
                {
                        if(!form.chk[i].checked)
                                form.chk[i].checked = true;
                        else
                                form.chk[i].checked = false;
                }
            }
        return false;
}
```

使用 alldel()或 overdel()选中复选框后，即可调用 del()函数来实现删除功能。del()函数首先使用 for 循环，将被选中的复选框的 value 值取出并存成数组，然后根据数组生成 URL，并使用 xmlhttp 对象调用这个 URL，当处理完毕后，根据返回值弹出提示或刷新本页。在 shopcar.js 脚本文件中，del()函数代码如下：

```
/*删除记录*/
function del(form){
        if(!window.confirm('是否要删除数据??')){              //删除确认

        }else{
                var leng = form.chk.length;                    //获取长度
                if(leng==undefined){
                        if(!form.chk.checked){
                                alert('请选取要删除数据!');
                        }else{
                                rd = form.chk.value;           //获取删除的 ID
                                var url = 'delshop.php?rd='+rd;   //定义执行的文件
                                xmlhttp.open("GET",url,true);     //GET 方法传递
                                xmlhttp.onreadystatechange = delnow;
                                xmlhttp.send(null);               //发送请求
                        }
                }else{
```





```
            var rd=new Array();
            var j = 0;
            for( var i = 0; i < leng; i++)
            {
                if(form.chk[i].checked){
                    rd[j++] = form.chk[i].value;
                }
            }
            if(rd == ''){
                alert('请选取要删除数据!');
            }else{
                var url = "delshop.php?rd="+rd;
                xmlhttp.open("GET",url,true);
                xmlhttp.onreadystatechange = delnow;
                xmlhttp.send(null);
            }
        }
    }
    return false;
}
function delnow(){                    //服务器响应
    if(xmlhttp.readyState == 4){
        if(xmlhttp.status == 200){
            var msg = xmlhttp.responseText;
            if(msg != '1'){
                alert('删除失败'+msg);
            }else{
                alert('删除成功');
                location=('?page=shopcar');
            }
        }
    }
}
```

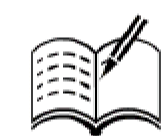
5.6.5 保存购物车

当用户希望保存更改后的商品数量时，可以单击“继续购物”按钮，将触发 onClick()事件，调用 conshop()函数保存数据。该函数有一个参数，就是当前表单的名称。在 conshop()函数内，根据复选框和购买数量文本域，生成两个数组 fst 和 snd，分别保存商品 ID 和商品数量。

这里要注意，两个数组的值是要相互对应的，如商品 1 的 ID 保存到 fst[1]中，那么商品 1 的数量就要保存到 snd[1]中，然后根据这两个数组生成一个 URL，使用 xmlhttprequest 对象调用 URL，最后根据回传信息作出相应的判断。conshop()函数存储于 js\shopcar.js 文件中，其代码如下。

```
//更改商品数量
function conshop(form){
```

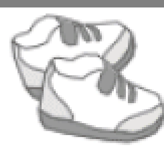





```
		var n_pre = 'cnum';
		var lang = form.chk.length;
		if(lang == undefined){
			var fst = form.chk.value;
			var snd = form.cnum0.value;
		}else{
			var fst= new Array();
			var snd = new Array();
			for(var i = 0; i < lang; i++){
				var nm = n_pre+i.toString();
				var stmp = document.getElementById(nm).value;
				if(stmp	== '' || isNaN(stmp)){
					alert('不允许为空、必须为数字');
					document.getElementById(nm).select();
					return false;
				}
				snd[i] = stmp;
				var ftmp = form.chk[i].value;
				fst[i] = ftmp;
			}
		}
		var url = 'changecar.php?fst='+fst+'&snd='+snd;		//定义执行文件
		xmlhttp.open("GET",url,true);					//通过 get 方法提交
		xmlhttp.onreadystatechange = updatecar;
		xmlhttp.send(null);							//发送请求
	}
	function updatecar(){
		if(xmlhttp.readyState == 4){					//判断是否执行成功
			var msg = xmlhttp.responseText;
			if(msg == '1'){
				location='index.php';				//页面跳转
			}else{
				alert('操作失败'+msg);
			}
		}
	}
```

在 conshop()函数中调用的 changecar.php 页为数据处理页，该页将商品 ID 和购买数量进行重新排列，并保存到 shopping 字段内。changecar.php 页面代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );		//设置文件编码格式
require("system/system.inc.php");					//包含配置文件
$sql = "select id,shopping from tb_user where name = '".$_SESSION['member']."'";
$rst = $admindb->ExecSQL($sql,$conn);
$reback = '0';
$changecar = array();
if(isset($_GET['fst']) && isset($_GET['snd'])){		//判断传递参数是否设置
	$fst = $_GET['fst'];							//获取参数值
```


```
        $snd = $_GET['snd'];
        $farr = explode(',',$fst);
        $sarr = explode(',',$snd);
        $upcar = array();
        for($i = 0; $i < count($farr); $i++){
            $upcar[$i] = $farr[$i].','.$sarr[$i];
        }
        if(count($farr) > 1){                                    //定义更新语句
        $update = "update tb_user set shopping='".implode('@',$upcar)."' where name = '".$_SESSION['member']."'";
        }else{
            $update = "update tb_user set shopping='".$upcar[0]."' where name = '".$_SESSION['member']."'";
        }
        $shop = $admindb->ExecSQL($update,$conn);            //执行更新操作
        if($shop){
            $reback = 1;
        }else{
            $reback = 2;
        }
    }
    echo $reback;
    ?>
```

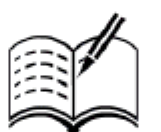
5.7 技术提炼

5.7.1 数据库连接、管理和分页类文件

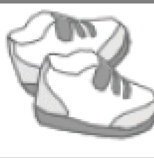
在数据库连接、管理和分页类文件中定义三个类：ConDB 数据库连接类，实现通过 PDO 连接 MySQL 数据库；AdminDB 数据库管理类，使用 PDO 类库中的方法执行对数据库中数据的查询、添加、更新和删除操作；SepPage 分页类，用于对商城中的数据进行分页输出。system.class.inc.php 文件存储于根目录下的 system 文件夹下，其具体代码如下：

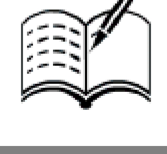
```
    <?php
    //数据库连接类
    class ConnDB{
        var $dbtype;
        var $host;
        var $user;
        var $pwd;
        var $dbname;
        function ConnDB($dbtype,$host,$user,$pwd,$dbname){            //构造方法
            $this->dbtype=$dbtype;
        $this->host=$host;
            $this->user=$user;
            $this->pwd=$pwd;
```






```
                $this->dbname=$dbname;
            }
        function GetConnId(){                              //实现数据库的连接并返回连接对象
            if($this->dbtype=="mysql" || $this->dbtype=="mssql"){          //判断数据库类型
                $dsn="$this->dbtype:host=$this->host;dbname=$this->dbname";
                }else{
                    $dsn="$this->dbtype:dbname=$this->dbname";
                }
                try {
        $conn = new PDO($dsn, $this->user, $this->pwd); //初始化 PDO 对象，就是创建数据
库连接对象$pdo
                    $conn->query("set names utf8");                    //设置编码格式
                return $conn;
                } catch (PDOException $e) {
                    die ("Error!: " . $e->getMessage() . "<br/>");
                }
        }
    }
    //数据库管理类
    class AdminDB{
        function ExecSQL($sqlstr,$conn){
            $sqltype=strtolower(substr(trim($sqlstr),0,6));           //对 SQL 语句进行截取
            $rs=$conn->prepare($sqlstr);                               //准备查询语句
            $rs->execute();                                   //执行查询语句，并返回结果集
            if($sqltype=="select"){
                $array=$rs->fetchAll(PDO::FETCH_ASSOC); //获取结果集中的所有数据
                if(count($array)==0 || $rs==false)
                    return false;
                else
                    return $array;
            }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
                if($rs)
                    return true;
                else
                    return false;
            }
        }
    }
    //分页类
    class SepPage{
        var $rs;
        var $pagesize;
        var $nowpage;
        var $array;
        var $conn;
        var $sqlstr;
        function ShowData($sqlstr,$conn,$pagesize,$nowpage){           //定义方法
            if(!isset($nowpage) || $nowpage=="")                       //判断变量值是否为空
                $this->nowpage=1;                         //定义首页从第一条记录开始读取
```

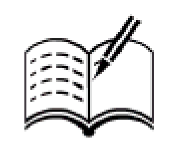





Note

```
            else
                $this->nowpage=$nowpage;
            $this->pagesize=$pagesize;                        //定义每页输出的记录数
            $this->conn=$conn;                                //连接数据库返回的标识
            $this->sqlstr=$sqlstr;                            //执行的查询语句
            $offset=($this->nowpage-1)*$this->pagesize;
            $sql=$this->sqlstr." limit $offset, $this->pagesize";
            $result=$this->conn->prepare($sql);               //准备查询语句
            $result->execute();                               //执行查询语句，并返回结果集
            $this->array=$result->fetchAll(PDO::FETCH_ASSOC); //获取结果集中的所有数据
            if(count($this->array)==0 || $this->array==false)
                return false;
            else
                return $this->array;
        }
        function ShowPage($contentname,$utits,$anothersearchstr,$anothersearchstrs,$class){
            $str="";
            $res=$this->conn->prepare($this->sqlstr);    //准备查询语句
            $res->execute();                             //执行查询语句，并返回结果集
            $this->array=$res->fetchAll(PDO::FETCH_ASSOC);//获取结果集中的所有数据
            $record=count($this->array);                      //统计记录总数
            $pagecount=ceil($record/$this->pagesize);         //计算共有几页
            $str.=$contentname." ".$record." ".$utits." 每页 
    ".$this->pagesize." ".$utits."  第  ".$this->nowpage."  页 / 共  ".$pagecount." 页";
            $str.="    ";
            if($this->nowpage!=1)
                $str.="<a href=".$_SERVER['PHP_SELF']."
    ?page=1&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class."> 首页</a>";
            else
                $str.="<font color='#555555'>首页</font>";
            $str.=" ";
            if($this->nowpage!=1)
                $str.="<a href=".$_SERVER['PHP_SELF']."?page=".($this->nowpage-1)."
    &page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs."  class=".$class."> 上一页</a>";
            else
                $str.="<font color='#555555'>上一页</font>";
            $str.=" ";
            if($this->nowpage!=$pagecount)
                $str.="<a href=".$_SERVER['PHP_SELF']."?page=".($this->nowpage+1)."
    &page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">下一页</a>";
            else
                $str.="<font color='#555555'>下一页</font>";
            $str.=" ";
            if($this->nowpage!=$pagecount)
                $str.="<a href=".$_SERVER['PHP_SELF']."?page=".$pagecount."
```





Note

```
&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">尾页</a>";
            else
                $str.="<font color='#555555'>尾页</font>";
        if(count($this->array)==0 || $this->array==false)
            return "无数据！ ";
        else
            return $str;
        }
}
?>
```

5.7.2　Smarty 模板配置类文件

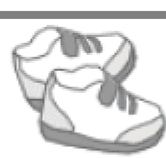
在 Smarty 模板配置类文件中，对 Smarty 模板文件、编译文件、配置文件等文件路径进行设置。system.smarty.inc.php 文件依然存储于项目根目录的 system 文件夹下，其代码如下：

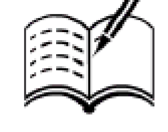
```
<?php
require("libs/Smarty.class.php");                        //包含模板文件
class SmartyProject extends   Smarty{                    //定义类，继承模板类
    function SmartyProject(){                            //定义方法
        $this->template_dir = "./system/templates/";     //指定模板文件存储位置
        $this->compile_dir = "./system/templates_c/";    //指定编译文件存储位置
        $this->config_dir = "./system/configs/";         //指定配置文件存储位置
        $this->cache_dir = "./system/cache/";            //指定缓存文件存储位置
    }
}
?>
```

5.7.3　执行类的实例化文件

在 system.inc.php 文件中，通过 require 语句包含 system.smarty.inc.php 和 system.class.inc.php 文件，执行类的实例化操作，并定义返回对象。完成数据库连接类的实例化后，调用其中的 GetConnId()方法连接数据库。system.inc.php 文件存储于 system 文件夹下，其代码如下：

```
<?php
require("system.smarty.inc.php");                    //包含 Smarty 配置类
require("system.class.inc.php");                     //包含数据库连接和操作类
$connobj=new ConnDB("mysql","localhost","root","111","db_business"); //数据库连接类实例化
$conn=$connobj->GetConnId();                         //执行连接操作，返回连接标识
$admindb=new AdminDB();                              //数据库操作类实例化
$seppage=new SepPage();                              //分页类实例化
$usefun=new UseFun();                                //使用常用函数对类进行实例化
$smarty=new SmartyProject();                         //调用 Smarty 模板
function unhtml($params){
   extract($params);
```



Note

```
        $text=$content;
        global $usefun;
        return $usefun->UnHtml($text);
    }
    $smarty->register_function("unhtml","unhtml");            //注册模板函数
    ?>
```

5.7.4 Smarty 模板页中的框架技术

在电子商务网站首页中应用 switch 语句与 Smarty 模板中的内建函数 include()设计一个框架页面，实现不同功能模块在首页中的展示。

（1）switch 语句

在 PHP 动态文件中使用，根据超链接传递的值，包含不同的功能模块。

（2）include 标签

在 Smarty 模板页中使用，在当前模板页中包含其他模板文件。其语法格式如下：

```
{include file="file_name " assign=" " var="    "}
```

参数说明如下。

☑ file：指定包含模板文件的名称。

☑ assign：指定一个变量，保存包含模板的输出。

☑ var：传递给待包含模板的本地参数，只在待包含模板中有效。

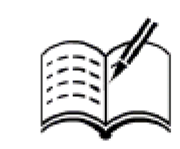
5.7.5 Ajax 无刷新验证技术

通过 Ajax 无刷新技术可以对用户名进行验证操作，以此来避免用户在进行会员注册时出现重名的问题。其具体的应用方法如下：

（1）Ajax 无刷新技术验证用户名是否被占用，调用的是 check.js 脚本中的 chkname 方法。其关键代码如下：

```
/*form 为传入的表单名称，本段代码为 register 表单*/
function chkname(form){
    /*如果 name 文本域的信息为空，名为 name1 的 div 标签显示如下信息*/
    if(form.name.value==""){
        name1.innerHMRL="<font color=#FF0000>请输入用户名！</font>";
    }else{
        /*否则，获取文本域的值*/
        var user = form.name.value;
        /*生成 url 链接，将 user 的值传到 chkname.php 页进行判断*/
        var url = "chkname.php?user="+user;
        /*使用 xmlhttprequest 技术运行页面*/
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
                /*根据不同的返回值，在 div 标签中输出不同信息*/
                var msg = xmlhttp.responseText;
```





Note

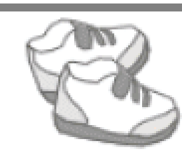
```
                        if(msg == '3'){
                            name1.innerHMRL="<font color=#FF0000>用户名被占用！</font>";
                            return false;
                        }else if(msg == '2'){
                            name1.innerHMRL="<font color=green>恭喜您，可以注册!</font>";
                        /*如果用户名正确，则将隐藏域的值改为"yes"*/
                            form.c_name.value = "yes";
                        }else{
                            name1.innerHMRL="<font color=green>未知错误</font>";
                        }
                    }
                }
                xmlhttp.send(null);
        }
}
```

在该函数中调用了 chkname.php 页，该页在会员登录时也会被调用，所以这里分两种情况，有密码和无密码。无密码为注册验证，当没有返回结果时，说明该用户名可用；而有密码为登录验证，和无密码相反，只有查询记录存在时，才允许登录，并将用户名和用户 ID 存储到 SESSION 中。chkname.php 页面代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require("system/system.inc.php");                                       //包含配置文件
$reback = '0';
$sql = "select * from tb_user where name='".$_GET['user']."'";
if(isset($_GET['password'])){
    $sql .= " and password = '".md5($_GET['password'])."'";
}
$rst = $admindb->ExecSQL($sql,$conn);
if($rst){
    /*    登录    */
    if($rst[0]['isfreeze'] != 0){
        $reback = '3';
    }else{
        $_SESSION['member'] = $rst[0]['name'];
        $_SESSION['id'] = $rst[0]['id'];
        $reback = '2';
    }
}else{
    $reback = '1';
}
echo $reback;
?>
```

（2）创建 yzm.php 文件，通过 GD2 函数库生成验证码。其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编
```




```
码格式
srand((double)microtime()*1000000);                    //生成随机数
$im=imagecreate(60,30);                                //创建画布
$black=imagecolorallocate($im,0,0,0);                  //定义背景
$white=imagecolorallocate($im,255,255,255);            //定义背景
$gray=imagecolorallocate($im,200,200,200);             //定义背景
imagefill($im,0,0,$gray);                              //填充颜色
for($i=0;$i<4;$i++){                                   //定义 4 位随机数
 $str=mt_rand(3,20);                          //定义随机字符所在位置的 Y 坐标
 $size=mt_rand(5,8);                          //定义随机字符的字体
 $authnum=substr($_GET['num'],$i,1);          //获取超链接中传递的验证码
 imagestring($im,$size,(2+$i*15),$str,$authnum,imagecolorallocate($im,rand(0,130),rand(0,130),rand(0,130)));
}                                             //水平输出字符串
for($i=0;$i<200;$i++){                        //执行 for 循环，为验证码添加模糊背景
  $randcolor=imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255));    //创建背景
  imagesetpixel($im,rand()%70,rand()%30,$randcolor);                  //绘制单一元素
}
imagepng($im);                                                        //生成 png 图像
imagedestroy($im);                                                    //销毁图像
?>
```

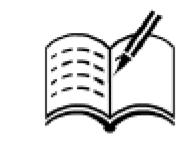
5.7.6　分页技术

商品展示功能实现的关键就是如何从数据库中读取商品信息，如何完成数据的分页显示。首先，包含类的实例化文件；然后，判断分页变量 page 的值是否存在；接着，定义 SQL 语句，对查询结果进行降幂排列，并且设置每页显示 3 条记录；最后，调用分页类中的方法完成数据的分页读取和输出。allnom.php 文件的关键代码如下：

```
<?php
include_once("system/system.inc.php");        //包含类的实例化文件
if(isset($_GET['page'])){                     //判断当前页变量的值
        $page=$_GET['page'];
}else{
        $page=1;
}
$rst1=$seppage->ShowData("select * from tb_commo where isnom=1 order by isnom,id desc",$conn,3,$page);//调用分页类方法
$smarty->assign("nomarr",$rst1);
$smarty->assign('rst1_page',$seppage->ShowPage("产品","个",$_GET['page_type'],",","a")); //输出分页超链接
$smarty->assign('title','推荐商品');
?>
```

最后，定义模板文件 allnom.tpl，通过 section 语句循环输出存储在模板变量中的数据，并且输出分页超链接。





5.7.7　购物车中商品添加技术

购物车功能实现最关键的部分就是如何将商品添加到购物车，如果不能完成商品的添加，那么购物车中的其他操作也就没有任何意义。

（1）在商品展示模块中，单击商品对应的“购买”按钮，将商品放到购物车中，并进入到购物车页面。单击“购买”按钮，调用 buycommo()函数，购买商品的 ID 是该函数的唯一参数。在 buycommo()函数中通过 xmlhttp 对象调用 chklogin.php 文件，并根据回传值作出相应处理。buycommo()函数代码如下：

```
/*
*添加商品，同时检查用户是否登录、商品是否重复等
*/
function buycommo(key){
	/*根据商品 ID，生成 url*/
	var url = "chklogin.php?key="+key;
	/*使用 xmlhttp 对象调用 chklogin.php 页*/
	xmlhttp.open("GET",url,true);
	xmlhttp.onreadystatechange = function(){
		if(xmlhttp.readyState == 4){
				var msg = xmlhttp.responseText;
				/*用户没有登录*/
				if(msg == '2'){
					alert('请您先登录');
					return false;
				}else if(msg == '3'){
				/*商品已添加*/
					alert('该商品已添加');
					return false;
				}else{
				/*显示购物车*/
					location='index.php?page=shopcar';
				}
			}
	}
	xmlhttp.send(null);
```

在 chklogin.php 文件中将商品添加到购物车中。chklogin.php 页代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
require("system/system.inc.php");                             //包含配置文件
/**
	*	1 表示添加成功
	*	2 表示用户没有登录
	*	3 表示商品已添加过
	*	4 表示添加时出现错误
	*	5 表示没有商品添加
*/
$reback = '0';
```

```
if(empty($_SESSION['member'])){
    $reback = '2';
}else{
    $key = $_GET['key'];
    if($key == ''){
        $reback = '5';
    }else{
        $boo = false;
        $sqls = "select id,shopping from tb_user where name = '".$_SESSION ['member']."'";
        $shopcont = $admindb->ExecSQL($sqls,$conn);
        if(!empty($shopcont[0]['shopping'])){
            $arr = explode('@',$shopcont[0]['shopping']);
            foreach($arr as $value){
                $arrtmp = explode(',',$value);
                if($key == $arrtmp[0]){
                    $reback = '3';
                    $boo = true;
                    break;
                }
            }
            if($boo == false){
                $shopcont[0]['shopping'] .= '@'.$key.',1';
    $update = "update tb_user set shopping='".$shopcont[0]['shopping']."' where name = '".$_SESSION['member']."'";
                $shop = $admindb->ExecSQL($update,$conn);
                if($shop){
                    $reback = 1;
                }else{
                    $reback = '4';
                }
            }
        }else{
            $tmparr = $key.",1";
            $updates = "update tb_user set shopping='".$tmparr."' where name = '".$_SESSION['member']."'";
            $result = $admindb->ExecSQL($updates,$conn);
            if($result){
                $reback = 1;
            }else{
                $reback = '4';
            }
        }
    }
}
echo $reback;
?>
```

通过分析上述代码可知，shopping 字段保存的是购物车中的商品信息。一条商品信息包括两部分，即商品 ID 和商品数量，其中商品数量默认为 1。两部分之间使用逗号“,”分隔，如果添加多个商品，则每个商品之间使用“@”分隔。

成功完成商品的添加操作后，即可进入到购物车页面，执行其他的操作。

第6章

在线支付模块

（自学视频、源程序：配套资源\mr\6\）

随着Internet的发展，电子商务已经成为21世纪网络发展的主流，网上购物和网上支付已经成为一种时尚。目前国内企业和个人纷纷加入到阿里巴巴等一系列大型电子商务网站中，通过网络进行B2B、B2C和C2C的网上交易。在进行网上交易的同时，付款方式也在随着网络的发展发生着变化，其中在线支付方式凭借其快捷、方便的特点，已经得到越来越多用户的认可。为了帮助广大PHP爱好者开发出功能更加完善的电子商务类网站，在本模块中将介绍如何通过支付宝实现电子商务网站的在线支付功能，以及订单生成和订单打印技术的实现方法。通过本章的学习，读者能够学到以下内容。

- 动态生成订单号
- 订单打印
- 支付宝支付
- 订单查询
- 统计购物车中的商品金额
- 向购物车中添加商品
- 清空购物车
- 通过PDO操作MySQL数据库
- 利用Smarty模板实现网页动静分离





Note

6.1 在线支付模块概述

6.1.1 功能概述

在线支付与购物车的作用相同，都是电子商务网站必不可少的一部分。本模块在购物车模块的基础上，增加了填写收货人信息、生成订单、订单预览、订单打印和提交订单等辅助功能，最终实现了支付宝在线支付功能。在线支付模块的功能结构如图 6.1 所示。

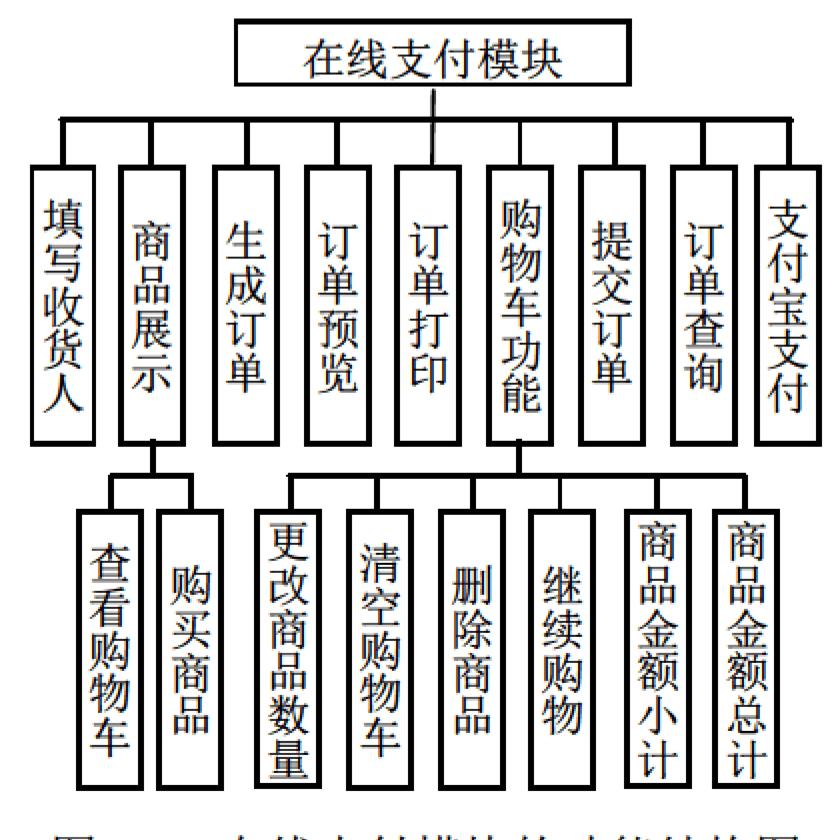


图 6.1 在线支付模块的功能结构图

6.1.2 在线支付操作流程

在线支付模块的操作流程与购物车十分相似，唯一区别就是增加了支付宝支付的功能。其操作流程如图 6.2 所示。

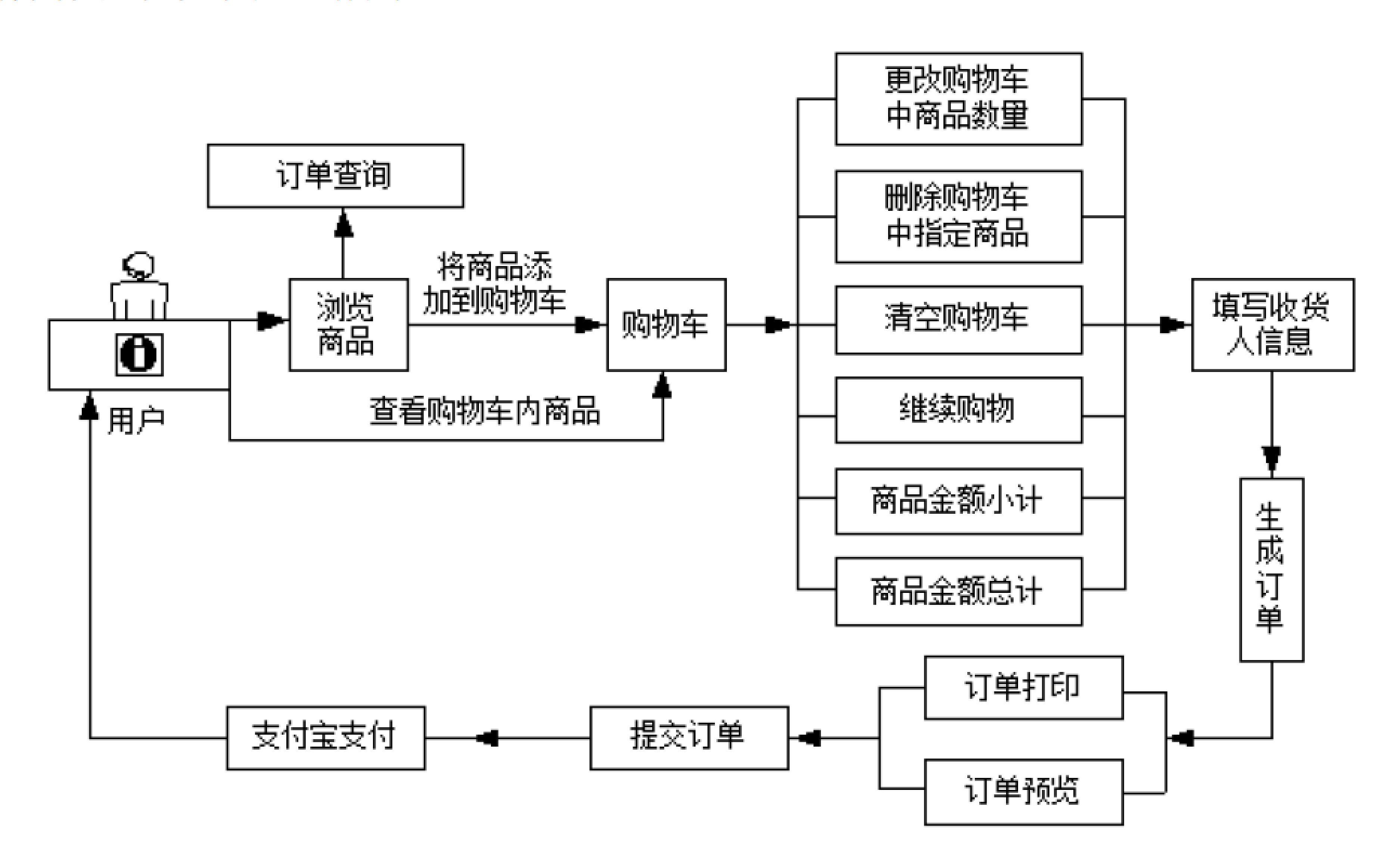
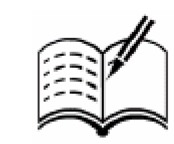


图 6.2 在线支付的操作流程





Note

6.1.3　程序预览

在线支付模块是电子商务网站收银台中的一个子功能，为了让读者对本模块有个初步的了解和认识，下面列出几个典型功能的页面。

明日购物商城网站填写收货人信息的页面如图 6.3 所示，可在其中填写收货人姓名、电话、联系地址等信息。

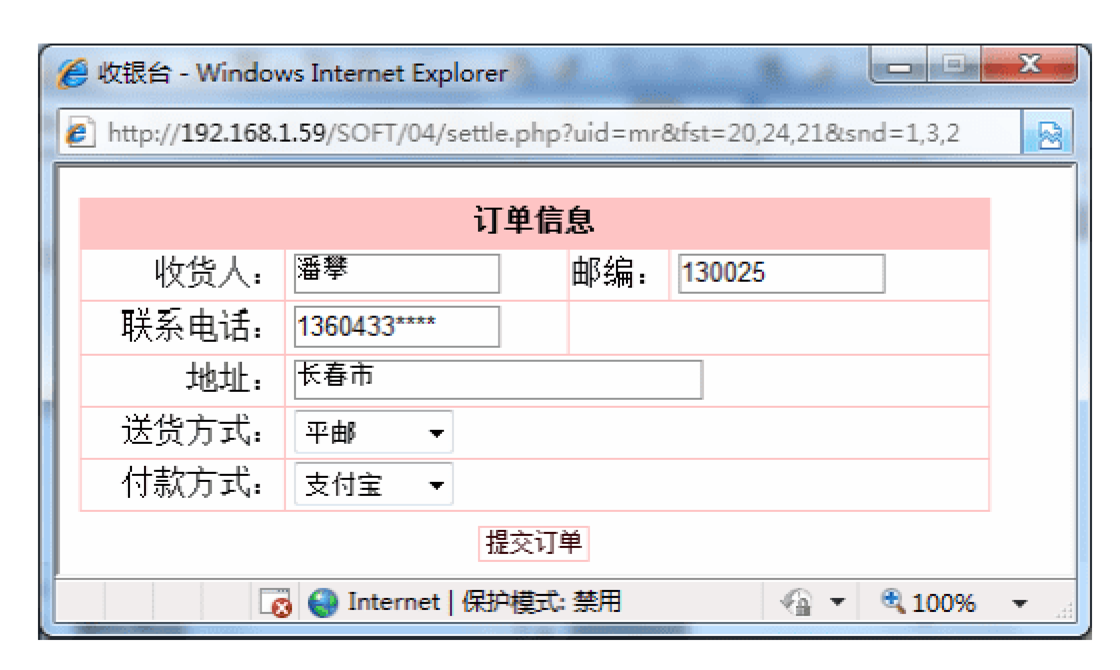


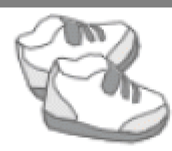
图 6.3　收货人信息添加页面

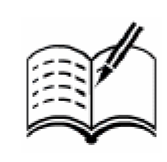
收货人信息填写完毕之后，就是提交订单，并且生成订单号，进而完成订单的预览、打印以及在线支付等操作。生成订单信息的页面如图 6.4 所示。



图 6.4　生成订单信息

在图 6.4 所示页面中单击“打印预览”按钮，将进入打印预览页面，如图 6.5 所示。单击“我要打印”按钮，将执行订单的打印操作，如图 6.6 所示。





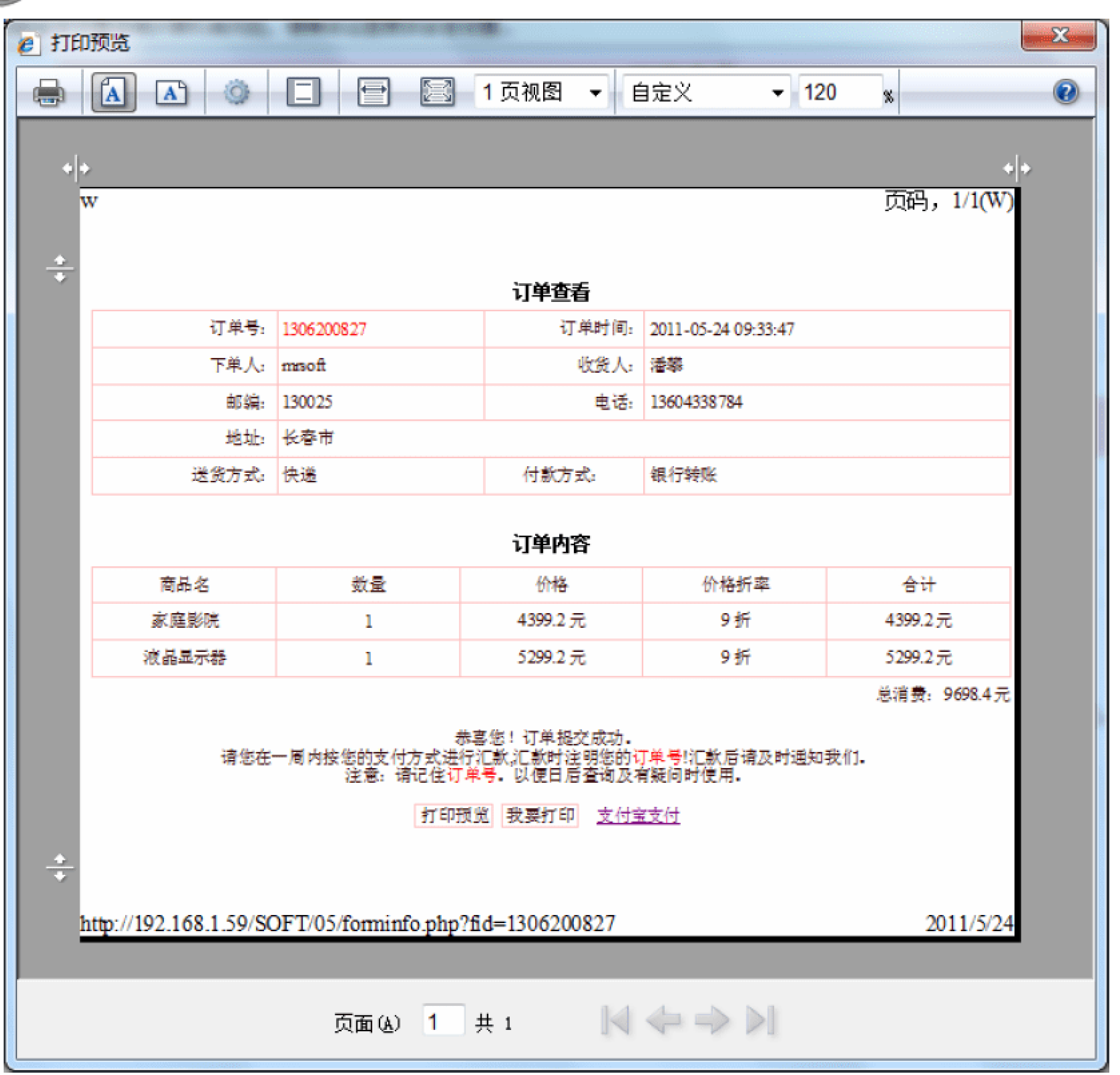


图 6.5　打印预览页面

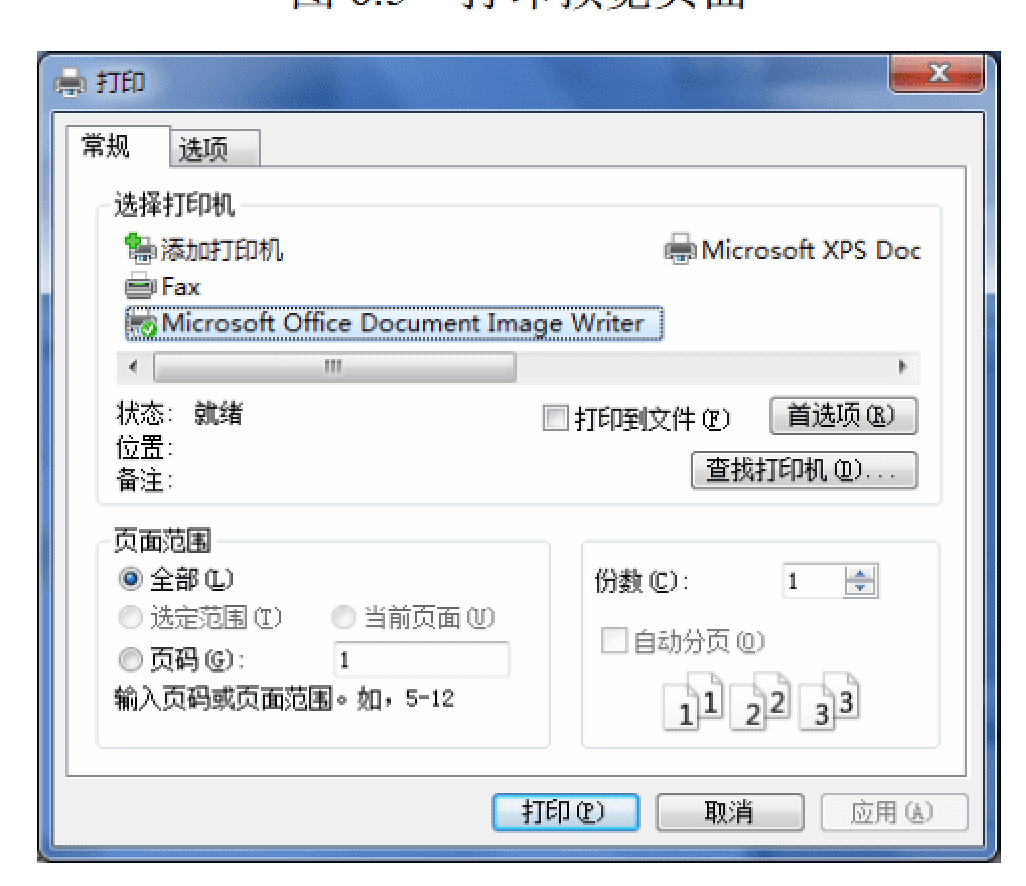


图 6.6　订单打印

6.2　数据库设计

在线支付模块与购物车模块同属于电子商务网站中的一个子功能模块，二者使用的数据库是统一的，都是 MySQL 数据库。在线支付模块与购物车模块的不同之处就是调用的是不同的数据表。



6.2.1　创建数据库

再来看一下电子商务网站所使用的数据表情况，如图 6.7 所示。



服务器: localhost ▸ 数据库: db_business

结构 | SQL | 搜索 | 查询 | 导出 | Import | 操作 | 权限 | 删除

表	操作	记录数	类型	整理	说明
tb_admin		2	MyISAM	utf8_unicode_ci	管理员表
tb_class		13	MyISAM	utf8_unicode_ci	商品类别表
tb_commo		4	MyISAM	utf8_unicode_ci	商品信息表
tb_form		17	MyISAM	utf8_unicode_ci	商品订单表
tb_links		2	MyISAM	utf8_unicode_ci	友情链接表
tb_opinion		0	MyISAM	utf8_unicode_ci	商品留言表
tb_public		4	MyISAM	utf8_unicode_ci	商品公告表
tb_user		6	MyISAM	utf8_unicode_ci	会员信息表
8 个表	总计	48	MyISAM	utf8_general_ci	

图 6.7　电子商务网站数据表

6.2.2　创建数据表

下面了解一下与在线支付有直接关系的数据表的结构。

☑　tb_commo（商品信息表）

商品信息表主要用于存储商品的相关信息，其结构如图 6.8 所示。

服务器: localhost ▸ 数据库: db_business ▸ 表 : tb_commo

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	utf8_unicode_ci		否			商品名称
pics	varchar(200)	utf8_unicode_ci		否	pics/null.jpg		商品图片
info	mediumtext	utf8_unicode_ci		否			商品介绍
addtime	date			否			添加时间
area	varchar(50)	utf8_unicode_ci		否			商品产地
model	varchar(50)	utf8_unicode_ci		否			商品型号
class	varchar(50)	utf8_unicode_ci		否			商品类型
brand	varchar(50)	utf8_unicode_ci		否			商品名称
stocks	int(4)			否	1		商品库存
sell	int(4)			否	0		销售量
m_price	float			否			市场价格
v_price	float			否			会员价格
fold	float			否	9		打折率
isnew	int(1)			否	1		是否新品
isnom	int(1)			否	0		是否推荐

图 6.8　商品信息表结构

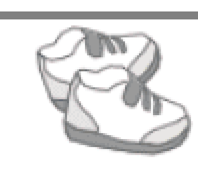
☑　tb_form（订单信息表）

订单信息表主要用于存储订单的相关信息，其结构如图 6.9 所示。

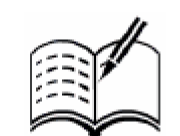
服务器: localhost ▸ 数据库: db_business ▸ 表 : tb_form

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
formid	varchar(125)	utf8_unicode_ci		否			订单号
commo_id	varchar(100)	utf8_unicode_ci		否			商品ID
commo_name	varchar(50)	utf8_unicode_ci		否			商品名称
commo_num	varchar(100)	utf8_unicode_ci		否			商品数量
agoprice	varchar(50)	utf8_unicode_ci		否			商品单价
fold	varchar(50)	utf8_unicode_ci		否			打折率
total	varchar(50)	utf8_unicode_ci		否			消费总额
vendee	varchar(50)	utf8_unicode_ci		否			消费者
taker	varchar(50)	utf8_unicode_ci		否			收货人姓名
address	varchar(200)	utf8_unicode_ci		否			收货地址
tel	varchar(20)	utf8_unicode_ci		否			联系电话
code	varchar(10)	utf8_unicode_ci		否			邮编
pay_method	varchar(20)	utf8_unicode_ci		否			付款方式
del_method	varchar(20)	utf8_unicode_ci		否			邮寄方式
formtime	timestamp			否	CURRENT_TIMESTAMP		订单日期
state	int(1)			否			订单状态

图 6.9　订单信息表结构



此外，还有管理员表、商品公告表、会员信息表、友情链接表和商品留言表等，限于篇幅，这里不再一一介绍，读者可参见配书光盘中的数据库文件。



Note

6.3　会员信息模块设计

6.3.1　会员信息模块概述

用户登录后，即可看到会员信息模块。在这里，可以进行查看或修改个人信息及密码、查看购物车和安全退出等操作。本节只对会员信息模块中的“会员中心”和“安全退出”进行讲解。会员信息模块的运行效果如图 6.10 所示。

mr信息（不可更改信息）	
会员编号：	2
会员名称：	mr
密保问题：	123456
密保答案：	123456
注册时间：	2010-12-06 05:14:10
消费总额：	0
mr信息（可更改信息）	
真实姓名：	潘攀 *
身份证号：	220322821103653 *
移动电话：	13604338784 *
固定电话：	0431-84972266 *
Email：	pp@sina.com
QQ号：	346827366
邮编：	130025
地址：	长春市 *
	修改　重置

图 6.10　会员信息页面

6.3.2　会员中心

当用户登录成功后，将看到会员中心超链接。当单击该超链接时，会回传给当前页一个 page 值，当前页根据这个 page 值来载入 member.php 文件。

1. 创建 PHP 页面

在此首先创建 PHP 页面。因为该模块中的模板需要使用数据库中的数据及一些动态信息，这些都需要在 PHP 页中先行获取及处理，然后再传给模板页。会员中心页面的代码参考 6.6 节。

2. 创建模板页

该模块包括查看信息模板及修改密码模板，都存储在 member.tpl 模板文件中。关键代码如下：

```
<link rel="stylesheet" href="css/member.css" />
```



```
<script language="javascript" src="js/member.js"></script>
{if $check=="find" }
<p  align="left">{$smarty.session.member}&gt;&gt;&gt;<a  href='?page=hyzx'  id="mem">查看信息</a>&gt;&gt;
<a href='?page=hyzx&action=modify' id="mem">修改密码</a></p>
<table  id="member" width="300" border="0" cellpadding="0" cellspacing="0">
<form  id="member"  name="member"  method="post"  action="modify_pwd_chk.php" onSubmit="return pwd(member)">
   <tr>
<td height="25" colspan="2" align="center" valign="middle" id="first"><font color="#f0f0f0">修改密码</font></td>
   </tr>
…//省略了部分代码
   <tr>
      <td height="30" colspan="2" align="center" valign="middle">
<input id="enter" name="enter" type="submit" value="修改" /></td>
   </tr>
</form>
</table>
{else}
<p  align="left">{$smarty.session.member}&gt;&gt;&gt;<a  href='?page=hyzx'  id="mem">查看信息</a>&gt;
<a href='?page=hyzx&action=modify' id="mem">修改密码</a></p>
{section name=pwd_id loop=$pwdarr}
<table id='member' width="500" border="0" cellpadding="0" cellspacing="0">
<form  id="member"  name="member"  method="post"  action="modify_info_chk.php" onSubmit="return mem(member)">
   <tr>
      <td height="25" colspan="2" align="center" valign="middle" id="first">
<font color="#f0f0f0">{$pwdarr[pwd_id].name}信息（不可更改信息）</font></td>
   </tr>
   <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left"> 会员编号：</td>
      <td height="25" align="left" valign="middle" id="right">  {$pwdarr[pwd_id].id} </td>
   </tr>
…//省略了部分代码
   <tr>
      <td height="30" colspan="2" align="center" valign="middle">
<input name="enter" type="submit" id="enter" value="修改" />    
<input name="reset" type="reset" id="reset" value="重置" /></td>
   </tr>
</form>
</table>
{/section}
{/if}
```

3. 创建脚本文件

该模块的脚本文件（存储于 06\js\member.js 中）和用户注册模块类似，都是对信息的



合法性进行验证，如信息是否为空、是否符合规范等，这里不再赘述。

4. 创建处理页



Note

当信息验证通过后，系统将跳转到处理页 modify_info_chk.php 进行信息处理。信息处理页的代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require("system/system.inc.php");                                       //包含配置文件
$sql = "update tb_user set realname='".$_POST['realname']."',card='".$_POST['card']."',tel='".$_POST['tel']."',
phone='".$_POST['phone']."',Email='".$_POST['email']."',QQ='".$_POST['qq']."',code='".$_POST['code']."',
address='".$_POST['address']."' where id = '".$_POST['userid']."'";
$arr = $admindb->ExecSQL($sql,$conn);
if($arr)
      echo "<script>alert('修改成功');location=('index.php');</script>";
else
      echo "<script>alert('修改失败');history.go(-1);</script>";
?>
```

6.3.3 安全退出

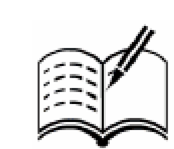
当用户需要离开网站时，可以单击“安全退出”超链接来调用 logout()函数；当用户确认退出后，则跳转到 logout 页面，销毁 session 并回到首页。安全退出所涉及的页面及代码如下：

```
function logout(){
      if(confirm("确定要退出登录吗？ ")){          //输出选择框，用户可以单击“确认”或“取消”按钮
            window.open('logout.php','_parent','',false);   //如果用户确认退出，则打开logout.php页
      }else
       return false;
}
```

```
<?php
      session_start();
      header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
      session_destroy();
      echo '<script>alert(\'用户已安全退出!\');location=(\'index.php\');</script>';
?>
```



6.4　收银台模块设计



6.4.1　收银台模块概述

当用户停止浏览商品准备结账时，可以单击购物车页面中的“去收银台”按钮，将触发 onClick()事件，调用 formset()函数显示订单页面。当用户提交订单后，系统将订单保存到数据表 tb_form 中，同时清空购物车，并显示订单信息提醒用户记录订单号。当货款发出后，还可以对订单进行查询。收银台页面的运行效果如图 6.11 所示。

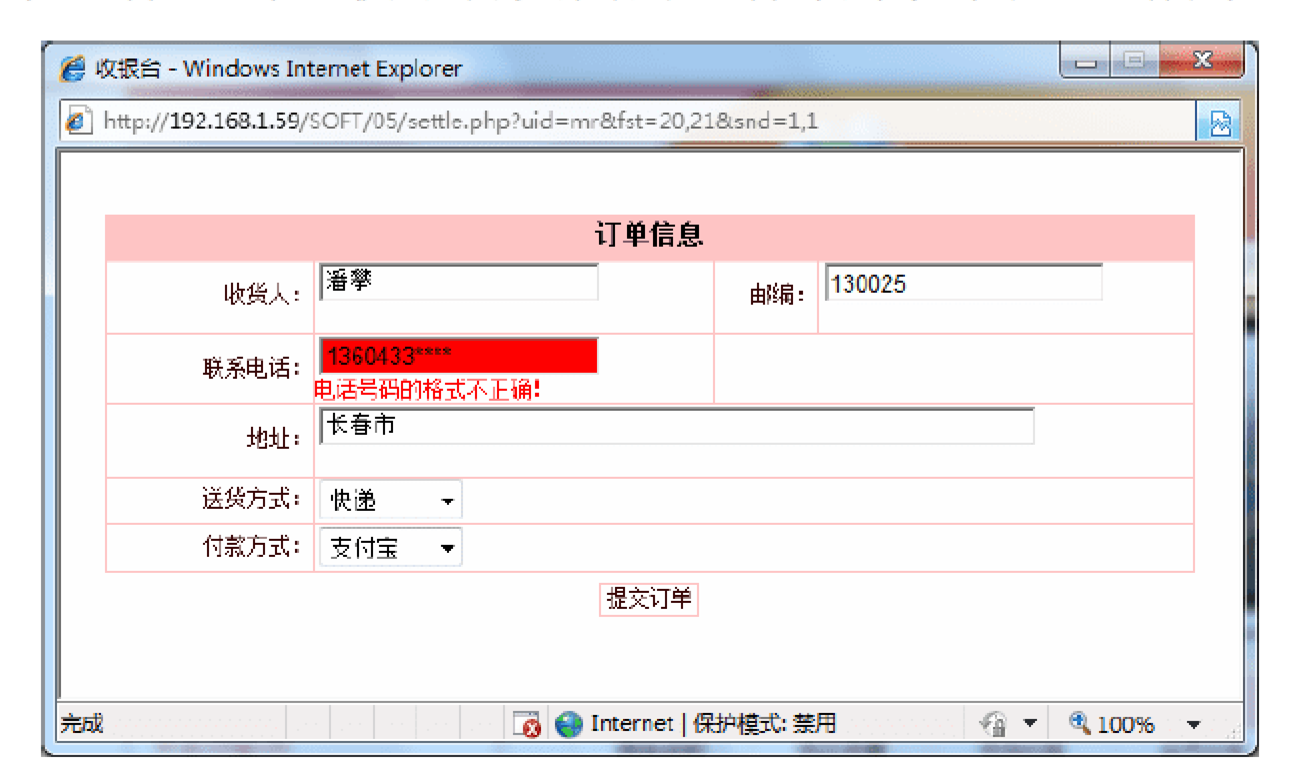


图 6.11　收银台页面运行效果

本节所涉及的页面包括显示订单（formset()函数）、填写订单（settle.php、settle.tpl），处理订单（settle_chk.php）、生成订单（forminfo.php、forminfo.tpl）以及订单预览打印 5 部分。

6.4.2　显示订单

订单信息提交页面的输出由 formset()函数决定，它将商品信息加以整理，通过 open()方法打开 settle.php 页来显示订单，并将整理后的商品信息传递到 settle.php 文件中。formset()函数存储于 js\shopcar.js 文件中，其关键代码如下：

```
function formset(form){
var uid = form.uid.value;
var n_pre = 'cnum';                                    //数量
    var lang = form.chk.length;
    if(lang == undefined){
        var fst = form.chk.value;                      //商品 ID
        var snd = form.cnum0.value;                    //购买数量
    }else{
        var fst= new Array();
        var snd = new Array();
        for(var i = 0; i < lang; i++){
            var nm = n_pre+i.toString();
            var stmp = document.getElementById(nm).value;
```



```
                if(stmp    == '' || isNaN(stmp)){
                    alert('不允许为空、必须为数字');
                    document.getElementById(nm).select();
                    return false;
                }
                snd[i] = stmp;
                var ftmp = form.chk[i].value;
                fst[i] = ftmp;
            }
        }
        open('settle.php?uid='+uid+'&fst='+fst+'&snd='+snd,'_blank','width=500 height=450',false);
    }
```

说明：

因为open()方法使用了_blank参数来打开一个新的页面，session值传不过去，所以这里使用隐藏域来传递用户名。

6.4.3 填写订单

settle.php直接将接收的值传给settle.tpl模板，并载入settle.tpl模板。settle.php页面的代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );        //设置文件编码格式
require("system/system.inc.php");                           //包含配置文件
$fst = $_GET['fst'];
$snd = $_GET['snd'];
$uid = $_GET['uid'];
$smarty->assign('title','收银台');
$smarty->assign('fst',$fst);
$smarty->assign('snd',$snd);
$smarty->assign('uid',$uid);
$smarty->display('settle.tpl');
?>
```

在settle.tpl模板页，首先载入settle.js脚本文件，调用其中的chkreginfo()方法验证表单提交的值是否符合要求。然后，创建表单form_reg，提交收货人、联系电话、联系地址等信息，同时还将PHP页传递的变量以隐藏域的形式提交到处理页settle_chk.php。settle.tpl模板页的关键代码如下：

```
<form name="form_reg" method="post" action="settle_chk.php" onSubmit="return chkreginfo(form_reg,'all')">
    <tr>
        <td height="25" align="right" valign="middle" class="left">收货人：</td>
        <td width="221" height="25" align="left" valign="middle" class="center"> 
<input type="text" name="taker" size="20" onBlur="chkreginfo(form_reg,0)">
```

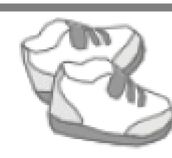


```
<div id="chknew_taker" style="color:#FF0000"></div>
        </td>
        <td width="56" align="right" valign="middle" class="center">邮编：</td>
        <td width="207" align="left" valign="middle" class="right"> 
<input type="text" name="code" size="20" onBlur="chkreginfo(form_reg,1)">
<div id="chknew_code" style="color:#FF0000"></div></td>
    </tr>
<!--省略了部分代码-->
    <tr>
        <td width="116" height="25" align="right" valign="middle" class="left"> 付款方式：</td>
      <td height="25" colspan="3" align="left" valign="middle" class="right"> <select id="pay" name="pay">
        <option value="银行转账">银行转账</option>
            <option value="邮局汇款">邮局汇款</option>
            <option value="支付宝">支付宝</option>
        </select>    </td>
    </tr>
    <tr>
        <td height="30" colspan="4" align="center" valign="middle">
<input id="enter" name="enter" type="submit" value="提交订单" class="btn" />
<input id="fst" name="fst" type="hidden" value="{$fst}" />
<input id="snd" name="snd" type="hidden" value="{$snd}" />
<input id="uid" name="uid" type="hidden" value="{$uid}" ></td>
    </tr>
</form>
```

6.4.4　处理订单

在处理页 settle_chk.php 中，首先获取表单中提交的数据；然后根据用户提交的商品信息，重新查找数据表 tb_commo，并从数据表中提取商品信息，保存到数组中，再将数组作为一条记录添加到 tb_form 表；数据添加成功的同时，处理页会根据 uid 找到该用户，将 shopping 字段清空；最后调用 forminfo.php 页来显示新添加的订单信息。settle_chk.php 页的代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
require("system/system.inc.php");                                   //包含配置文件
$sql="insert into tb_form(formid,commo_id,commo_name,commo_num,agoprice,fold,total,vendee,taker,address,
tel,code,pay_method,del_method,formtime,state)values(";
$formid=time();
$tmpid = explode(',',$_POST['fst']);
$tmpnm = explode(',',$_POST['snd']);
$number = count($tmpid);
$tmpna = array();
$tmpvp = array();
$tmpfd = array();
$tmptt = 0;
```

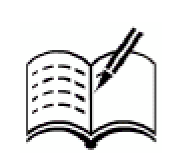



```
    if($number >1){
        for($i = 0; $i < $number; $i++){
            $tmpsql = "select name,v_price,fold from tb_commo where id = '".$tmpid[$i]."'";
            $tmprst = $admindb->ExecSQL($tmpsql,$conn);
            $tmpna[$i] = $tmprst[0]['name'];
            $tmpvp[$i] = $tmprst[0]['v_price'];
            $tmpfd[$i] = $tmprst[0]['fold'];
            $tmptt += $tmprst[0]['v_price'] * $tmpnm[$i];
            @$tmpsell = $tmprst[0]['sell'] + 1;
            $addsql = "update tb_commo set sell = '".$tmpsell."' where id = '".$tmpid[$i]."'";
            $addrst = $admindb->ExecSQL($addsql,$conn);
        }
        $sql.="'".$formid."','".$_POST['fst']."','".implode(',',$tmpna)."','".$_POST['snd']."','".implode(',',$tmpvp)."',
    '".implode(',',$tmpfd)."','".$tmptt."','".$_POST['uid']."'";
    }else if($number == 1){
        $tmpsql = "select name,v_price,fold from tb_commo where id = '".$tmpid[0]."'";
        $tmprst = $admindb->ExecSQL($tmpsql,$conn);
        $tmptt= $tmprst[0]['v_price'] * $tmpnm[0];
        @$tmpsell = $tmprst[0]['sell'] + 1;
        $addsql = "update tb_commo set sell = '".$tmpsell."' where id = '".$tmpid[0]."'";
        $addrst = $admindb->ExecSQL($addsql,$conn);
        $sql.="'".$formid."','".$_POST['fst']."','".$tmprst[0]['name']."','".$_POST['snd']."','".$tmprst[0]['v_price']."',
    '".$tmprst[0]['fold']."','".$tmptt."','".$_POST['uid']."'";
    }else{
        echo 'error';
        exit();
    }
    $sql.=",'".$_POST['taker']."','".$_POST['address']."','".$_POST['tel']."','".$_POST['code']."','".$_POST['pay']."',
    '".$_POST['del']."','".date("Y-m-d H:i:s")."',0)";
    $InsertSQL = $admindb->ExecSQL($sql,$conn);
    if(false == $InsertSQL){
        echo "<script>alert('购买失败');history.back;</script>";
    }else{
        $updsql = "update tb_user set consume='".$tmptt."',shopping='' where name = '".$_POST['uid']."'";
        $updrst = $admindb->ExecSQL($updsql,$conn);
        echo "<script>top.opener.location.reload();</script>";
    echo "<script>open('forminfo.php?fid=$formid','_blank','width=750 height=650',false);window.close();</script>";
    }
    ?>
```

6.4.5 生成订单

订单处理成功之后，将跳转到 forminfo.php 页输出生成的订单。订单生成的关键是





forminfo.php 和 forminfo.tpl。

在 forminfo.php 文件中，根据超链接传递的订单号查询订单表中的数据，并将查询结果赋给指定的模板变量，最终指定模板页 forminfo.tpl。其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include_once("system/system.inc.php");                              //包含类的实例化文件
$ddnumber=$_GET['fid'];                                             //获取订单号
$sql = "select * from tb_form where formid='".$_GET['fid']."'";     //定义 SQL 语句
$formarr = $admindb->ExecSQL($sql,$conn);                           //执行 SQL 语句
$commname = explode(',',$formarr[0]['commo_name']);                 //读取数据库中数据
$commnumber = explode(',',$formarr[0]['commo_num']);
$commagoprice = explode(',',$formarr[0]['agoprice']);               //读取数据库中数据
$commfold = explode(',',$formarr[0]['fold']);
$smarty->assign('formarr',$formarr[0]);                             //将数据赋给模板变量
$smarty->assign('commname',$commname);
$smarty->assign('commnumber',$commnumber);
$smarty->assign('commagoprice',$commagoprice);
$smarty->assign('commfold',$commfold);
if($formarr[0]["del_method"]=="平邮"){
        $yprice="200";
}else{
        $yprice="300";
}
…//省略部分代码
$smarty->assign('title','订单提交成功');
$smarty->display('forminfo.tpl');                                   //指定模板页
?>
```

在 forminfo.tpl 模板页中，获取模板变量传递的值，完成订单信息的输出，并且创建“我要打印”、“打印预览”按钮和“支付宝支付”超链接。其关键代码如下：

```
<table width="650" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="30" colspan="5" align="center" valign="middle" class="first">订单内容</td>
  </tr>
  <tr>
    <td width="100" height="25" align="center" valign="middle" class="left">商品名</td>
    <td width="100" height="25" align="center" valign="middle" class="center">数量</td>
    <td width="100" height="25" align="center" valign="middle" class="center">价格</td>
    <td width="100" height="25" align="center" valign="middle" class="center">价格折率</td>
    <td width="100" height="25" align="center" valign="middle" class="right">合计</td>
  </tr>
{foreach key=key item=item from=$commname}
  <tr>
    <td height="25" align="center" valign="middle" class="left">{$item}</td>
    <td height="25" align="center" valign="middle" class="center">{$commnumber[$key]}</td>
    <td height="25" align="center" valign="middle" class="center">{$commagoprice[$key]}
 元</td>
```

```
            <td align="center" valign="middle" class="center">{$commfold[$key]} 折</td>
        <td  align="center"  valign="middle"  class="right">{$commagoprice[$key]*$commnumber[$key]} 元</td>
        </tr>
    {/foreach}
    <tr>
            <td  colspan="5"  height="25"  align="right"  valign="middle">总消费：{$formarr.total} 元</td>
    </tr>
    </table>
    <input type="button" value="打印预览" onclick="printview()" class="btn" />  
    <input type="button" value="我要打印" onclick="prints()" class="btn" />   
    <!--<input type="button" value="我要打印" onclick="window.print()" class="btn" />-->
    <!--支付宝支付的接口操作，提交的数据-->
            <a href="{$link}">支付宝支付</a>
    <!--————————————————-->
```

说明：

本模块中的订单打印、打印预览和支付宝支付功能都是在 forminfo.php 和 forminfo.tpl 文件中完成的，有关这些内容将在后文中进行详细介绍。

6.4.6　订单预览、打印

在线支付模块中应用 WebBrowser 技术实现订单的打印和预览功能，运行效果如图 6.12 和图 6.13 所示。

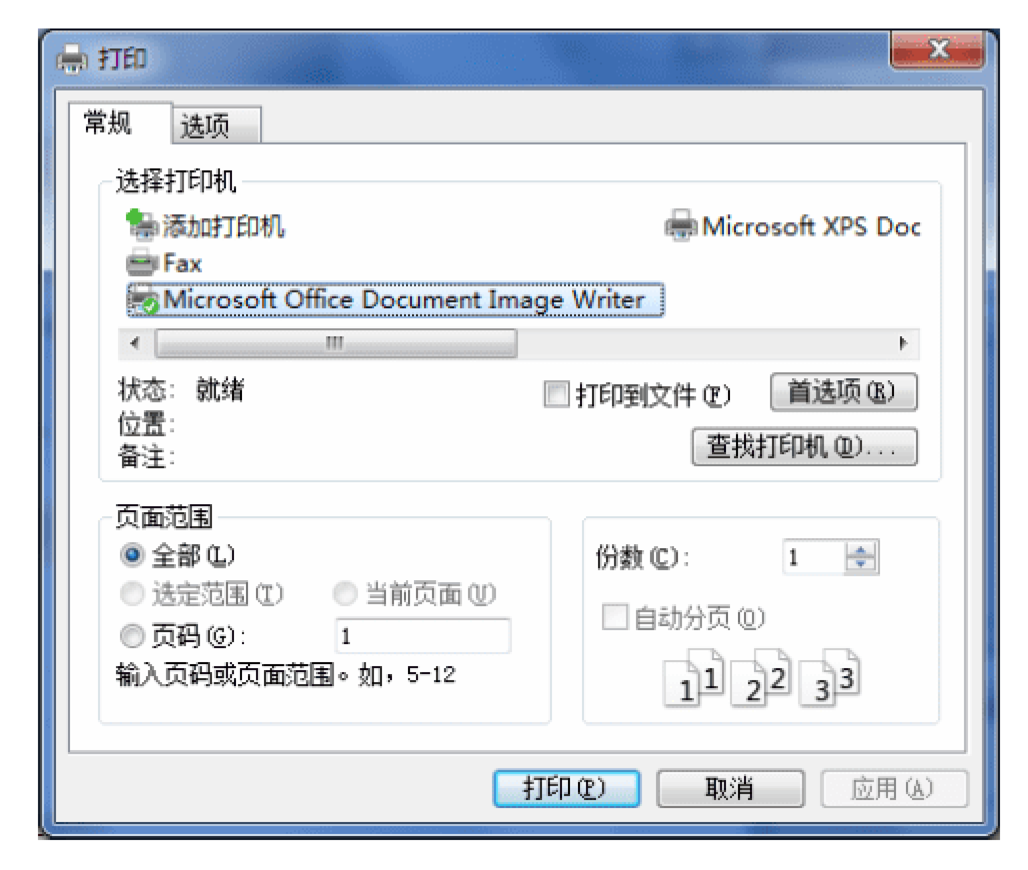




图 6.12　订单打印

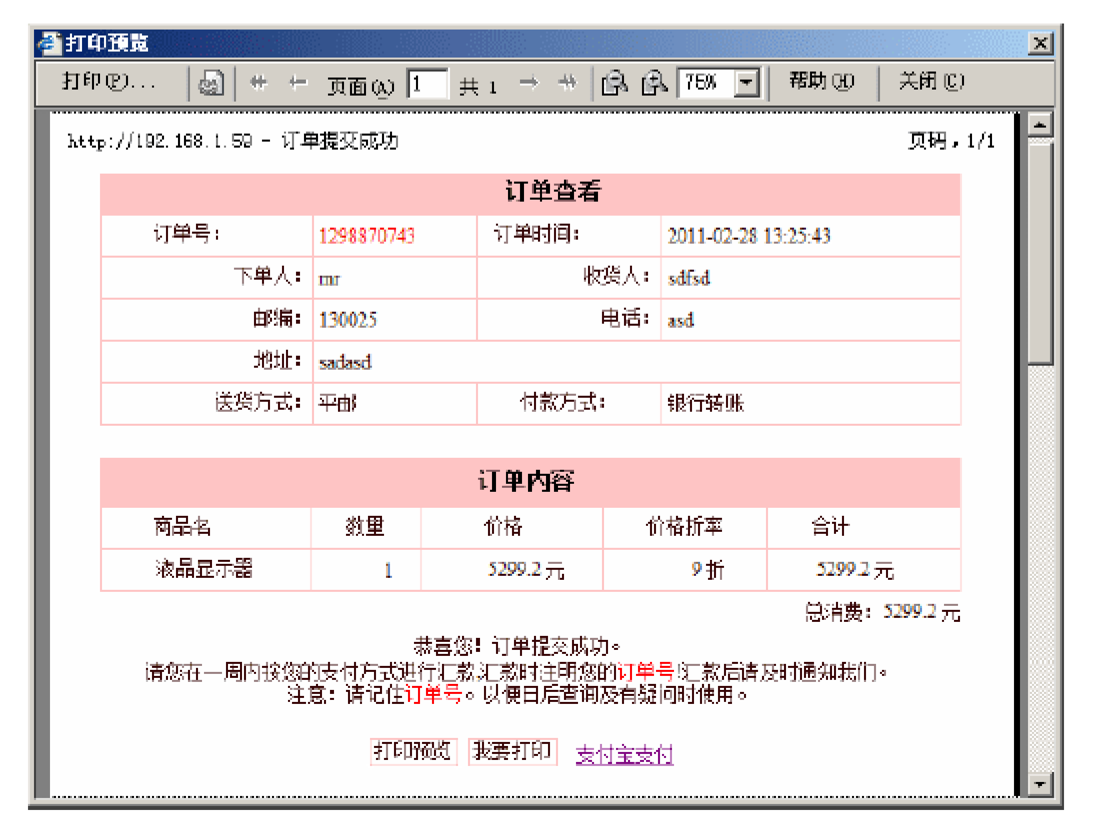


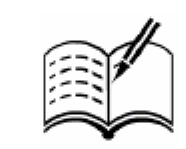
图 6.13　打印预览

本模块中的打印和预览功能在 forminfo.tpl 模板页中实现。首先，载入 print.js 脚本文件；然后，定义 object 标签；最后，创建打印预览和打印的提交按钮，并且通过 onClick()函数调用对应的方法完成相应操作。其关键代码如下：

```
    <script language="javascript" src="js/print.js"></script>
    <object ID='WebBrowser1'   CLASSID='CLSID:8856F961-340A-11D0-A96B- 00C04FD705A2'></object>
```



```
<input type="button" value="打印预览" onclick="printview()" class="btn" />
<input type="button" value="我要打印" onclick="prints()" class="btn" />   <!—
<input type="button" value="我要打印" onclick="window.print()" class="btn" />-->
```



print.js 脚本文件中封装了 printview()和 prints()两个方法，分别调用 WebBrowser 控件中的方法完成打印预览和打印操作。其代码如下：

```
function printview(){
    document.all.WebBrowser1.ExecWB(7,1);        //打印预览
    window.close();
}
function prints(){
    document.all.WebBrowser1.Execwb(6,1);        //打印
    window.close();
}
```

6.5　订单查询

6.5.1　订单查询功能概述

利用订单查询功能，可以根据提交的订单号从数据库中查询出指定订单的数据。其运行效果如图 6.14 所示。



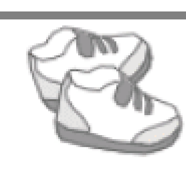
图 6.14　订单查询

6.5.2　订单查询功能的实现

订单查询功能由 4 个文件组成，分别是 queryform.php、queryform.tpl、queryform.js 和 query.php。具体实现过程如下：

（1）通过 queryform.php 文件动态 PHP 文件，用于指定模板页 queryform.tpl。

（2）利用 queryform.tpl 模板页创建订单查询的表单。载入 JS 脚本，通过 JS 脚本中





Note

的 queryrst()方法调用 query.php 完成订单的无刷新查询，将查询结果在 queryform.tpl 模板页中输出。queryform.tpl 模板页的关键代码如下：

```
<form id='queryform' name='queryform' method='post' action='#'>
  <tr>
    <td height='25' colspan='5' align='center' valign='middle' class='first'>查询订单</td>
  </tr>
  <tr>
    <td width='80' height='25' align='right' valign='middle' class='left'>查询用户：</td>
    <td width='130' height='25' align='left' valign='middle' class='center'> 
<input id='name' name='name' type='text' class='txt' /></td>

    <td width='100' height='25' align='right' valign='middle' class='center'>查询订单号:</td>
    <td width='130' height='25' align='left' valign='middle' class='right'> 
<input id='formid' name='formid' type='text' class='txt' /></td>
          <td width='100' align='center' valign='middle' class='center'>
<input id='enter' name='enter' type='button' value='查询' class='btn' onclick='return queryrst(queryform)'/></td>
  </tr>
  <tr>
    <td height='25' colspan='5' align='center' valign='middle'> </td>
    </tr>
</form>
```

（3）创建 queryform.js 脚本文件，定义 queryrst()和 showfm()方法，调用 query.php 文件完成订单的无刷新查询，并且将查询结果赋给 exam。其代码如下：

```
// JavaScript Document
function queryrst(form) {
    var name = form.name.value;                          //获取表单提交的值
    var formid = form.formid.value;                      //获取表单提交的值
    if(name == '' && formid == ''){                      //判断提交的值是否为空
        alert('用户或订单号至少有一个不能为空');
        form.name.focus();
        return false;
    }
    var url = "query.php?vendee="+name+"&formid="+formid;    //定义调用的 URL
    xmlhttp.open("GET",url,true);                            //执行调用操作
    xmlhttp.onreadystatechange = showfm;                     //回调函数
    xmlhttp.send(null);                                      //执行
}
function showfm(){
    if(xmlhttp.readyState == 4){                             //判断返回结果
        var msg = xmlhttp.responseText;
        if(msg == '0'){
                exam.innerHTML = '';
                alert('没有查询结果');
        }else{
                exam.innerHTML =  msg;                       //将查询结果赋给 exam
        }
```



```
        }
    }
```

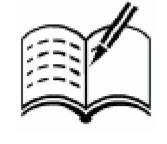
（4）创建 query.php 文件，根据超链接传递的参数值执行订单查询操作，并且返回查询结果。其代码如下：



```
<?php
header ( "Content-type: text/html; charset=UTF-8" );        //设置文件编码格式
include_once("system/system.inc.php");                      //载入类的实例化文件，加载对象
$vendee = $_GET['vendee'];                                  //获取超链接传递的值
$formid = $_GET['formid'];                                  //获取订单号
$reback = '0';
$tmp = "<table width='600' border='0' align='center' cellpadding='0' cellspacing='0'>
<tr><td height='25' colspan='7' align='center' valign='middle' class='first'>查询结果</td></tr>
<tr><td width='100' height='25' align='center' valign='middle' class='left'>订单号</td>
<td width='75' height='25' align='center' valign='middle' class='center'>订货会员</td>
<td width='75' height='25' align='center' valign='middle' class='center'>收货人</td>
<td width='100' height='25' align='center' valign='middle' class='center'>订单金额</td>
<td width='75' height='25' align='center' valign='middle' class='center'>付款方式</td>
<td width='75' height='25' align='center' valign='middle' class='center'>收款方式</td>
<td width='100' height='25' align='center' valign='middle' class='right'>订单状态</td></tr>";
$sql = "select id,formid,vendee,taker,total,pay_method,del_method,state from tb_form
where vendee = '".$vendee."' or formid = '".$formid."'";
$rst = $admindb->ExecSQL($sql,$conn);                       //执行查询语句
if($rst){
    foreach($rst as $value){                                //循环输出查询结果
            $tmp .= "<tr><td height=25 align=center valign=middle class=left>".$value
['formid']."</td>
<td align=center valign=middle class=center>".$value['vendee']."</td>
<td align=center valign=middle class=center>".$value['taker']."</td>
<td  align=center valign=middle class=center>".$value['total']."</td>
<td  align=center valign=middle class=center>".$value['pay_method']."</td>
<td align=center valign=middle class=center>".$value['del_method']."</td>
<td align=center valign=middle class=right>";
            switch ($value['state']){
                case 0:
                    $tmp .= '未作处理';
                    break;
                case 1:
                    $tmp .= '已付款';
                    break;
                case 2:
                    $tmp .= '已发货';
                    break;
                case 3:
                    $tmp .= '已收货';
                    break;
            }
            $tmp .= "</td></tr>";
        }
```



```
            $tmp .= "</table>";
            $reback = $tmp;
        }
        echo $reback;
?>
```



6.6 技 术 提 炼

6.6.1 收货人信息验证

为了保证网上交易的顺利完成，在处理收货人提交的信息时，必须对其填写的信息进行验证，以保证信息是合理、有效的，避免格式上的错误出现。这里在验证收货人提交的信息时应用的是 JavaScript 脚本和正则表达式。验证收货人信息的运行结果如图 6.15 所示。

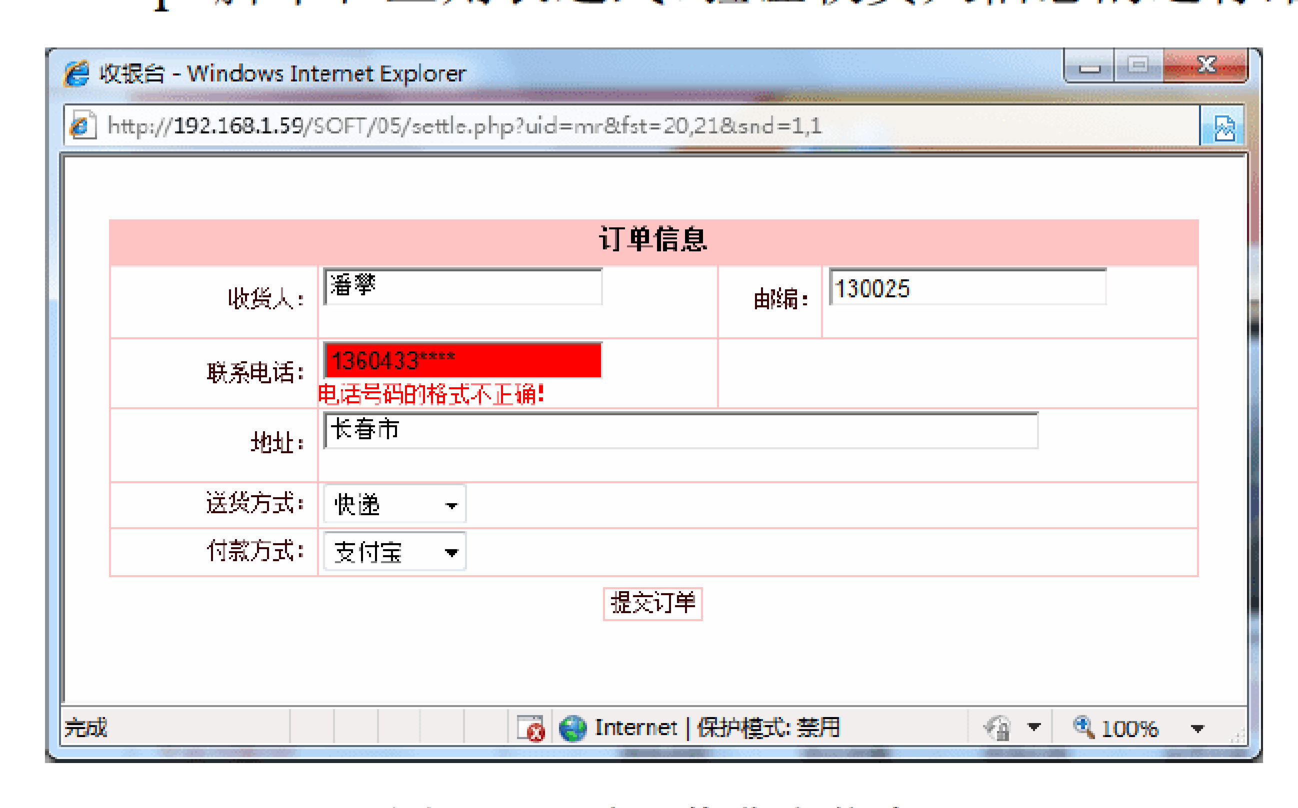


图 6.15 验证收货人信息

（1）验证手机号码格式

主要是通过 JavaScript 脚本的自定义函数，调用正则表达式，实现对手机号码的验证。编写判断手机号码格式的正则表达式如下：

```
/^13(\d{9})$|^15(\d{9})$/;
```

判断是否是以 13 或者 15 开头，判断是否是 11 位数字。判断手机号码格式是否正确的自定义函数的代码如下：

```
function checkregtel(regtel){
    var str=regtel;
    var Expression=/^13(\d{9})$|^15(\d{9})$/;    //判断手机号码的号段和格式是否正确
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
        return true;
    }else{
        return false;
    }
}
```

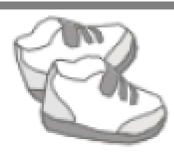


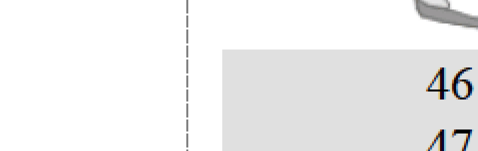


（2）验证收货人的其他信息

验证收货人的其他信息主要应用的是 JavaScript 脚本，通过 JavaScript 脚本中的自定义函数获取表单提交的数据，在自定义函数中判断数据是否合理，如果不合理则通过 div 标签直接输出错误提示。验证收货人的其他信息的代码如下：

```
function chkreginfo(form,mark,edit){
    if(mark==0 || mark=="all"){                //验证收货人
        if(form.taker.value==""){
            chknew_taker.innerHTML="请输入收货人！";
            form.taker.style.backgroundColor="#FF0000";
            return false;
        }else{
            chknew_taker.innerHTML="";
            form.taker.style.backgroundColor="#FFFFFF";
        }
    }
    if(mark==1 || mark=="all"){                //验证邮编
        if(form.code.value==""){
            chknew_code.innerHTML="请输入邮编！";
            form.code.style.backgroundColor="#FF0000";
            return false;
        }else if(isNaN(form.code.value)){
            chknew_code.innerHTML="邮编由数字组成！";
            form.code.style.backgroundColor="#FF0000";
            return false;
        }else if(form.code.value.length!=6){
            chknew_code.innerHTML="邮编由 6 位数字组成！";
            form.code.style.backgroundColor="#FF0000";
            return false;
        }else{
            chknew_code.innerHTML="";
            form.code.style.backgroundColor="#FFFFFF";
        }
    }
    if(mark==2 || mark=="all"){                //验证电话号码
        if(form.tel.value==""){
            chknew_tel.innerHTML="请输入电话号码！";
            form.tel.style.backgroundColor="#FF0000";
            return false;
        }else if(!checkregtel(form.tel.value)){
            chknew_tel.innerHTML="电话号码的格式不正确！";
            form.tel.style.backgroundColor="#FF0000";
            return false;
        }else if(isNaN(form.tel.value)){
            chknew_tel.innerHTML="电话号码由数字组成！";
            form.tel.style.backgroundColor="#FF0000";
            return false;
        }else{
            chknew_tel.innerHTML="";
            form.tel.style.backgroundColor="#FFFFFF";
```





```
                }
            }
            if(mark==3 || mark=="all"){         //验证联系地址
                if(form.address.value==""){
                    chknew_address.innerHTML="请输入联系地址！ ";
                    form.address.style.backgroundColor="#FF0000";
                    return false;
                }else{
                    chknew_address.innerHTML="";
                    form.address.style.backgroundColor="#FFFFFF";
                }
            }
        }
```

6.6.2 动态生成订单号

动态生成订单号应用的是 time()函数。time()函数用于获取当前的 UNIX 时间戳，返回值为从 UNIX 纪元（格林威治时间 1970 年 1 月 1 日 00:00:00）到当前时间的秒数。其语法格式如下：

```
int time ( void )
//time()函数没有参数，返回值为 UNIX 时间戳的整数值
```

这里就是以 time()函数返回的 UNIX 时间戳作为订单号的。在订单处理页 settle_chk.php 中，通过 time()获取 UNIX 时间戳的整数值，并且将其赋给$formid 变量作为订单号。其关键代码如下：

```
$formid=time();                              //生成订单号
```

6.6.3 WebBrowser 打印

在线支付模块中订单的打印和预览是非常重要的两个功能，实现这两个功能主要应用的是 WebBrowser 打印技术。打印预览的运行结果如图 6.16 所示。

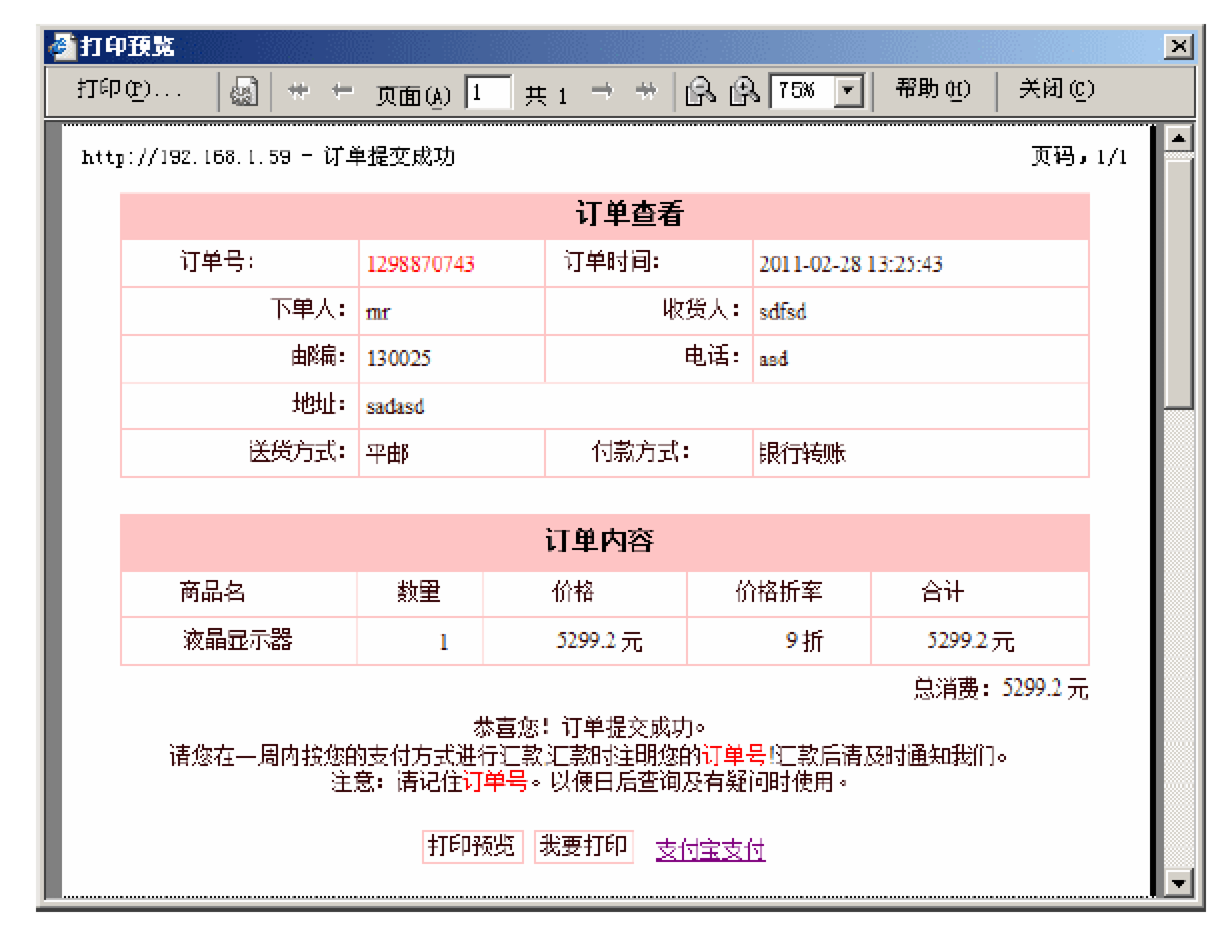
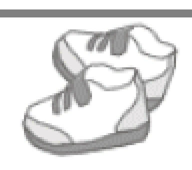


图 6.16　打印预览





Note

WebBrowser 是 IE 内置的浏览器控件，无需用户下载。其优点是客户端独立完成打印目标文档的生成，减轻服务器负荷；缺点是源文档的分析操作复杂，并且要对源文档中要打印的内容进行约束。

订单打印和预览主要应用 IE 内置的 WebBrowser 控件来实现，该控件的具体参数如表 6.1 所示。

表 6.1　IE 内置的 WebBrowser 控件的参数说明

参　数	说　明
document.all.WebBrowser.Execwb(7,1)	表示打印预览
document.all.WebBrowser.Execwb(6,1)	表示打印
document.all.WebBrowser.Execwb(6,6)	表示直接打印
document.all.WebBrowser.Execwb(8,1)	表示页面设置
document.all.WebBrowser.Execwb(1,1)	打开页面
document.all.WebBrowser.Execwb(2,1)	关闭所有打开的IE窗口
document.all.WebBrowser.Execwb(4,1)	保存网页
document.all.WebBrowser.Execwb(10,1)	查看页面属性
document.all.WebBrowser.Execwb(17,1)	全选
document.all.WebBrowser.Execwb(22,1)	刷新
document.all.WebBrowser.Execwb(45,1)	关闭窗体无提示

6.6.4　支付宝支付

本模块中通过支付宝完成在线支付。要使用支付宝进行在线支付，必须先在支付宝中注册一个账户，然后下载支付宝提供的接口文件，通过该文件实现电子商务网站与支付宝的交互。

在支付宝提供的接口文件中，提供了完整的实例程序，只要按照实例程序中的操作方法，将电子商务网站中指定的数据提交到支付宝提供的数组参数中，支付宝便会通过自定义的方法将数据提交到指定的网站中。具体实现过程如下。

（1）在 forminfo.php 文件中，根据从数据库中读取的数据，定义支付宝的数组参数。其关键代码如下：

```
require_once("alipay_service.php");
require_once("alipay_config.php");
$parameter = array(
"service" => "trade_create_by_buyer",   //交易类型，必填，实物交易＝trade_create_by_buyer（需要填写物流）
"partner" =>$partner,                    //合作商户号
"return_url" =>$return_url,              //同步返回
"notify_url" =>$notify_url,              //异步返回
"_input_charset" => $_input_charset,     //字符集，默认为 GBK
"subject" =>$formarr[0]["commo_name"],   //商品名称，必填
"body" => $formarr[0]["commo_num"],      //商品描述，必填
"out_trade_no" => $ddnumber,     //商品外部交易号，订单号，必填，每次测试都需修改
```




Note

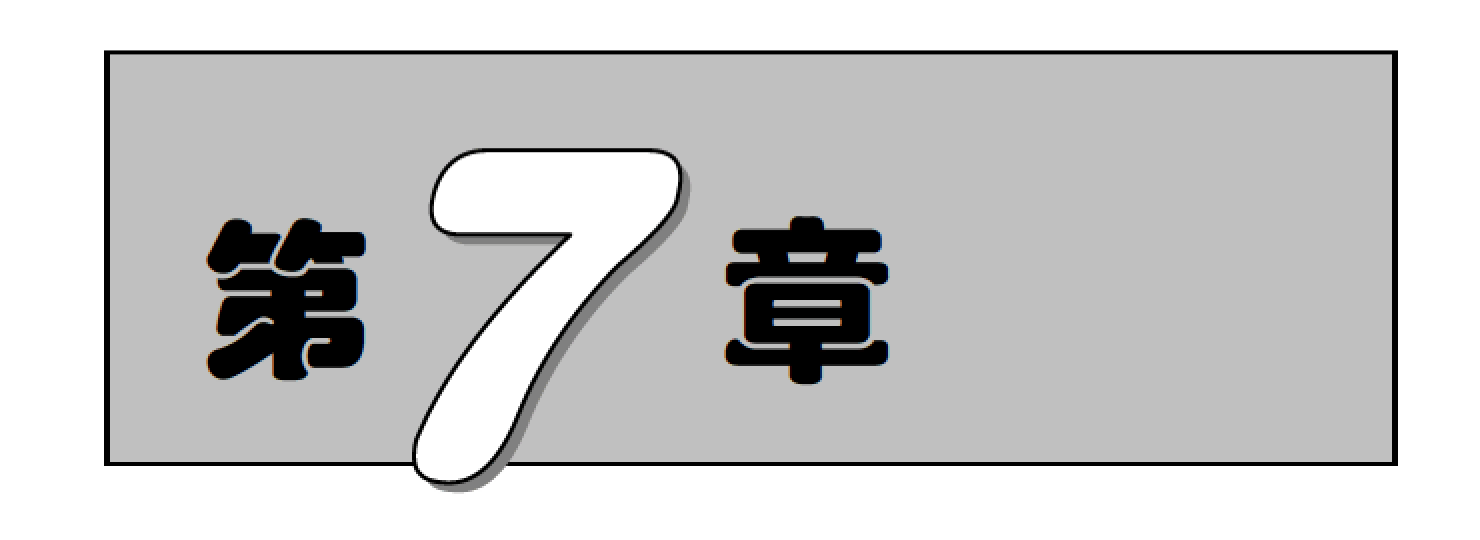
```
"logistics_fee"=>$yprice,                    //物流配送费用
"logistics_payment"=>'BUYER_PAY',  //物流配送费用付款方式：BUYER_PAY(买家支付)
"logistics_type"=>'EXPRESS',               // 物 流 配 送 方 式 ： POST( 平 邮 ) 、 EMS(EMS) 、
EXPRESS(其他快递)
"price" => $formarr[0]["total"],          //商品单价，必填
"payment_type"=>"1",                         //默认为 1，不需要修改
"quantity" => "1",                             //商品数量，必填
"show_url" => $show_url,                  //商品相关网站
"seller_email" => $seller_email          //卖家邮箱，必填
);
$alipay = new alipay_service($parameter,$security_code,$sign_type);
$link=$alipay->create_url();
$smarty->assign("link",$link);
```

（2）在 forminfo.tpl 模板页中，创建超链接，完成支付操作。其关键代码如下：

```
<!--支付宝支付的接口操作，提交的数据-->
        <a href="{$link}">支付宝支付</a>
<!--————————————————————————-->
```

说明：

在本模块中，只是介绍了 PHP 中电子商务网站与支付宝进行交互的方法，以及各种辅助功能的实现方法。至于支付宝提供的参数和设置都是个人虚拟的，真实的参数将在实际的运作中由支付宝提供。在实际的运作中，需要向支付宝申请一个账户，然后由支付宝提供相应的参数，就可以应用此功能了。



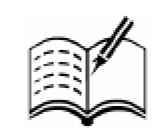
假日公告管理模块

（自学视频、源程序：配套资源\mr\7\）

上网浏览时，经常会遇到这样的情况，即打开网页后会弹出一个个小窗口，显示一些广告、公告或者假日问候等。在各大门户网站，这种现象十分普遍，尤其是企业网站更是如此。那么这是如何实现的呢？本章就来带领读者，亲自动手开发一个假日公告管理模块。通过本章的学习，读者能够学到以下内容。

- 数据库的连接与数据操作
- 通过超链接传递值
- 单文件上传
- 多图片上传
- 读取指定目录中的目录及文件
- JQuery 控制弹出公告信息





7.1 假日公告管理概述

7.1.1 模块概述

假日公告管理模块实现了发布及管理假日公告信息。当要发布假日公告时，在“假日公告发布”子模块中上传公告文件及图片即可，让用户及时了解公司的工作时间，并且可以在第一时间为用户送上假日祝福。假期过后，可以通过“假日公告取消”子模块，取消发布的假日公告，然后通过“假日公告管理”子模块将公告信息删除。

7.1.2 系统功能结构

假日公告管理模块的功能结构如图 7.1 所示。

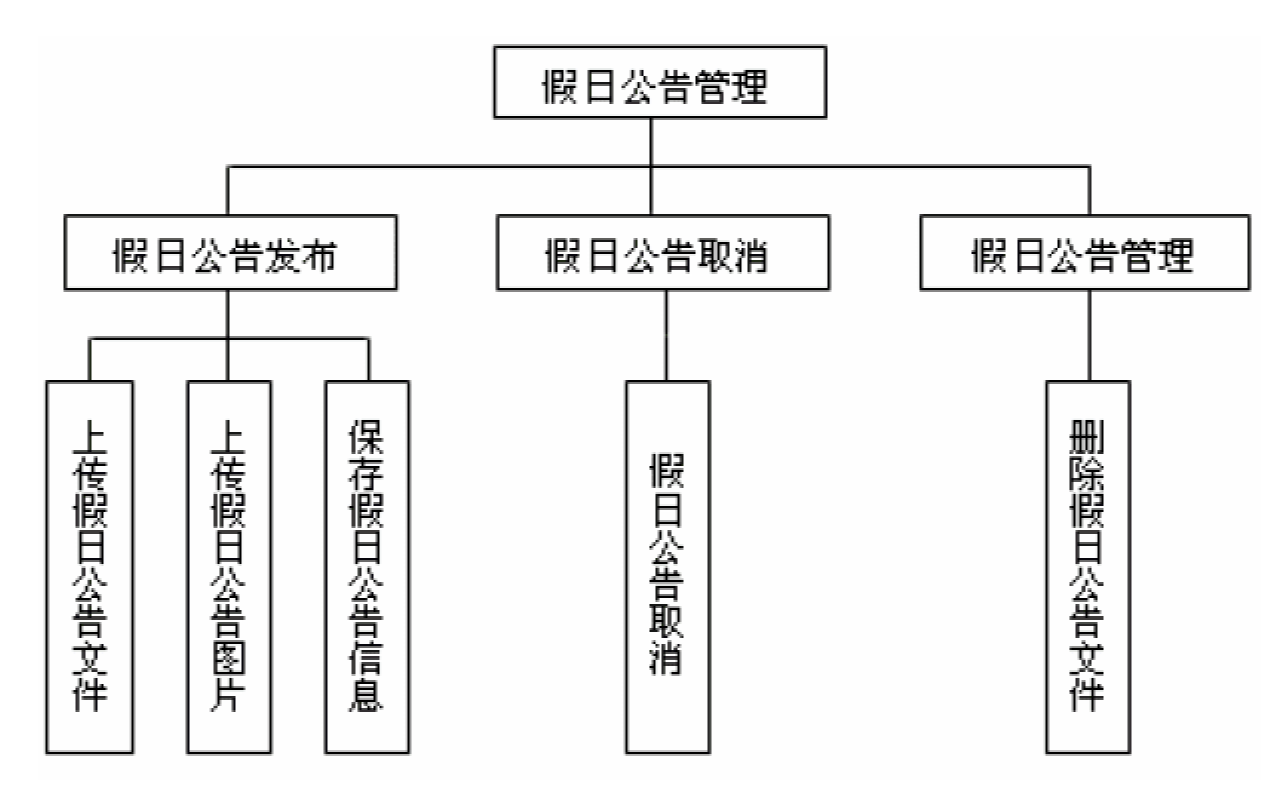


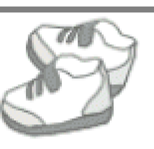
图 7.1　假日公告管理模块的功能结构图

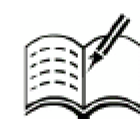
7.1.3 程序预览

为了让读者对假日公告管理模块有一个初步的了解和认识，下面列出其中一些功能模块的运行效果图，供大家参考。运行网站弹出公告信息页面，效果如图 7.2 所示。



图 7.2　运行网站时弹出的公告信息页面效果





Note

假日公告管理模块的主页运行效果如图 7.3 所示。



图 7.3　假日公告管理模块主页

登录页面运行效果如图 7.4 所示。

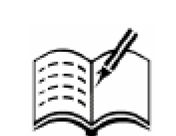


图 7.4　登录页面运行效果

假日公告发布页面运行效果如图 7.5 所示。



图 7.5　假日公告发布页面运行效果



假日公告取消页面运行效果如图 7.6 所示。



图 7.6　假日公告信息取消页面

假日公告管理页面运行效果如图 7.7 所示。



图 7.7　假日信息管理页面

7.2　数据库设计

7.2.1　创建数据库

通过对系统进行的需求分析、业务流程设计以及系统功能结构的确定，规划出系统中使用的数据库 db_jiari。数据库的结构及数据表的情况如图 7.8 所示。



服务器: localhost ▸ 数据库: db_jiari

表	操作	记录数	类型	整理	说明
tb_01		1	MyISAM	utf8_general_ci	假日公告信息表
tb_user		1	MyISAM	utf8_general_ci	管理员表
2 个表	总计	2	MyISAM	utf8_general_ci	

图 7.8　假日公告管理模块数据库与数据表



Note

7.2.2　创建数据表

下面看一下各个表的结构和说明。

☑　tb_01（假日公告信息表）

假日公告信息表用于存储发布的假日信息，其结构如图 7.9 所示。

服务器: localhost ▸ 数据库: db_jiari ▸ 表 : tb_01

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	utf8_unicode_ci		否			公告主题
perform	varchar(1)	utf8_general_ci		否			是否执行

图 7.9　假日公告信息表

☑　tb_user（管理员信息表）

管理员信息表主要用于存储管理员信息，其结构如图 7.10 所示。

服务器: localhost ▸ 数据库: db_jiari ▸ 表 : tb_user

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(10)	utf8_unicode_ci		否			管理员名称
pwd	varchar(10)	utf8_unicode_ci		否			管理员密码

图 7.10　管理员信息表

7.3　用 户 登 录

7.3.1　用户登录概述

用户登录模块是假日公告管理系统的门户，只有登录成功后，才可以进入系统主页，并进行假日发布、取消及删除等操作。登录模块运行效果如图 7.11 所示。



图 7.11　用户登录页面

7.3.2　用户登录功能实现过程

当用户输入登录信息后，首先判断用户名和密码是否为空，然后再判断用户名和密码





Note

是否正确，如果正确则登录成功，进入系统主页，否则将给出提示信息。

（1）判断用户名和密码是否为空，应用的是 checkit()自定义脚本函数。该函数存储于 user.php 文件中，其关键代码如下：

```
<script language="javascript">
function checkit(){                                    //自定义函数
        if(form1.name.value==""){                      //判断用户名是否为空
                alert("请输入用户名!");
                form1.name.select();
                return false;
        }
        if(form1.pwd.value==""){                       //判断密码是否为空
                alert("请输入密码!");
                form1.pwd.select();
                return false ;
        }
                return true;

}
</script>
```

（2）判断用户名和密码是否正确，如果正确则登录成功，否则将提示用户名和密码不正确。其操作是在 user.php 文件中完成的，代码如下：

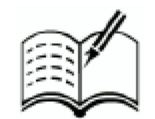
```
<?php
include("conn/conn.php");                                          //包含数据库连接文件
if(isset($_POST['name']) and $_POST['pwd']!=null){                 //判断用户名和密码是否为空
        $select=mysql_query("select * from tb_user
where name='".$_POST['name']."' and pwd='".$_POST['pwd']."'",$conn);   //执行查询操作
        if($row=mysql_num_rows($select)==1){
                $_SESSION['name']=$_POST['name'];
                echo "<script>alert('登录成功！');window.location.href='indexs.php';</script>";
        }else{
                echo "<script>alert('用户名和密码不正确！');window.location.href='user.php'; </script>";
        }
}
?>
…                                        //部分代码省略
<input type="image" name="imageField" onclick="return checkit();" src="images/user_05.gif" />
…                                        //部分代码省略
<input type="image" name="imageField2" onclick"return false;" src="images/user_07.gif" />
```

7.4 假日公告发布

7.4.1 假日公告发布概述

假日公告发布模块主要用于发布假日公告信息。进入假日公告发布页面，填写假日名





称，并选择假日文件及图片，然后单击“发布”按钮，即可完成公告信息发布。假日公告发布页面运行效果如图 7.12 所示。



图 7.12　假日公告发布页面

7.4.2　假日公告信息发布实现过程

（1）创建 form 表单，以 post 方式将数据提交到 promulgation_ok.php 文件中进行处理，并且设置 enctype="multipart/form-data"属性，同时添加假日名称、假日文件、假日图片和是否执行等表单元素。其关键代码如下：

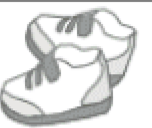
```
<form action="promulgation_ok.php" method="post" enctype="multipart/form-data" name="form1" id="form1">
    <table id="   01" width="800" height="600" border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td colspan="3"><a href="indexs.php">
<img src="images/manage  01.gif" alt="" width="800" height="201" border="0" /></a></td>
      </tr>
      <tr>
        <td><img src="images/manage_02.gif" width="140" height="251" alt="" /></td>
        <td width="523" height="251" align="center" valign="top" background="images/manage_03.gif">
<table width="416" height="232">
          <tr>
            <td width="89" height="30">假日名称：</td>
            <td width="315" align="left"><input name="name" type="text" size="40" /></td>
          </tr>
          <tr>
            <td height="30">假日文件：</td>
            <td align="left"><input name="files" type="file" size="31" /></td>
          </tr>
          <tr>
            <td height="30">假日图片 1：</td>
            <td align="left"><?php
```



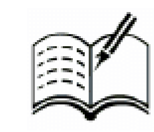
```
                for($a=0;$a<4;$a++){
            ?>
                        <input type="file" name="photo_<?php echo $a;?>" />
                        <?php
                }
            ?>
                    </td>
                </tr>
                 <tr>
                    <td height="30">是否执行：</td>
                    <td><label>
                        <input name="radio" type="radio" value="1" checked="checked" />
                        是
                        <input type="radio" name="radio" value="0" />
                        否</label></td>
                 </tr>
                 <tr>
                    <td height="30" colspan="2"><div align="center">
                        <input type="submit" name="Submit" value="发布" />
                        <input type="reset" name="Submit2" value="取消" />
                    </div></td>
                 </tr>
              </table></td>
              <td><img src="images/manage_04.gif" width="137" height="251" alt="" /></td>
          </tr>
          <tr>
              <td  colspan="3"><img  src="images/manage_05.gif"  width="800"  height="148" alt="" /></td>
          </tr>
        </table>
    </form>
```

（2）创建 promulgation_ok.php 文件，获取表单中提交的数据，完成假日公告信息的发布操作。其关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_POST['name']) and $_POST['Submit']=="发布"){
$insert=mysql_query("insert into tb_01(name,perform)
values('".$_POST['name']."','".$_POST['radio']."')",$conn);      //添加数据
        if(isset($_FILES['files']['name'])){                       //判断文件及图片是否存在
                if(!is_dir("./upfile/images")){                    //判断指定目录是否存在
                        mkdir("./upfile/images");                  //创建目录
                }
                $photo="upfile/".$_FILES['files']['name'];         //文件的存储路径和名称
                if(move_uploaded_file($_FILES['files']['tmp_name'],$photo)){ //执行上传
                        for($a=0;$a<4;$a++){                               //循环语句
                                $name=$_FILES['photo_'.$a];        //将变量的名称保存在变量中
                                $path='upfile/images/'.$name['name'];      //定义上传文件的路径
                                move_uploaded_file($name['tmp_name'],$path);
                        }
```



```
                echo "<script>alert('上传成功！');window.location.href='promulgation.php';</script>";
            }else{
                echo "<script>alert('上传失败！');window.location.href='promulgation.php';</script>";
            }
        }
    }else{
        echo "<script>alert('请选择上传文件！');window.location.href='promulgation.php';</script>";
    }
?>
```



Note

7.5　假日公告取消

7.5.1　假日公告取消概述

假期过后，在假日公告取消页面中单击假日名称后的“取消”超链接，即可取消该假日公告，并在数据库中删除该假日信息。假日公告取消页面运行效果如图 7.13 所示。



图 7.13　假日公告取消页面

7.5.2　浏览公告信息

打开假日公告取消页面，默认将显示所有假日信息。可以通过 select 查询语句查询信息表中的数据，通过 while 语句循环输出假日公告信息，并且为每条信息设置一个“取消”超链接，将信息的 ID 作为参数值传递到 update.php 文件中。cancel.php 的关键代码如下：

```
<?php
include("conn/conn.php");
$select=mysql_query("select * from tb_01",$conn);
while($array=mysql_fetch_array($select)){
?>
<tr>
    <td height="30"><?php echo $array['name'];?> </td>
```






```
            <td><div align="center">
                <a href="update.php?abrogation=<?php echo $array['id'];?>">取消</a>
            </div></td>
    /tr>
    <?php
    }
    ?>
```

7.5.3 取消假日公告

取消假日公告主要是通过超链接参数 abrogation 传递的值，在 update.php 页中执行 delete 语句删除信息，取消假日公告。update.php 的关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_GET['abrogation'])){
        $update=mysql_query("delete from tb_01 where id='".$_GET['abrogation']."'",$conn);
        if($update){
                echo "<script>alert('假日信息取消成功！');window.location.href='cancel.php';</script>";
        }else{
                echo "<script>alert('假日信息取消失败！');window.location.href='cancel.php';</script>";
        }
}
?>
```

7.6 假日公告管理

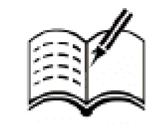
7.6.1 假日公告管理概述

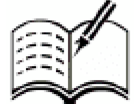
假日公告管理模块负责对上传的公告文件及图片进行管理，将过期的公告信息文件删除，以免造成空间浪费。假日公告管理模块运行效果如图 7.14 所示。



图 7.14 假日公告管理页面





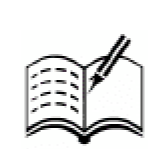


7.6.2　假日公告管理实现过程

（1）创建 manage.php 动态页，输出指定目录下的文件夹及文件，并实现“上级目录”的跳转。其关键代码如下：

```
<?php
if (! isset ( $_GET ['catalog'] ) || empty ( $_GET ['catalog'] ))
        $current_directory = getcwd ();                                    //获得脚本目录
else
        $current_directory = $_GET ['catalog'];
chdir ( iconv ( "utf-8", "gb2312", $current_directory ) );            //改变当前目录
echo "<span class='STYLE4'>
当前目录:</span><span class='STYLE2'>" . iconv ( "gb2312", "utf-8", getcwd () ) . "</span><br>";
$ml = opendir ( iconv ( "utf-8", "gb2312", $current_directory ) );        //打开目录
while ( $gain_directory = readdir ( $ml ) ) {                       //循环读取目录中的目录及文件
        echo "<tr><td align='center' valign='middle'>";
        if (is_dir ( $gain_directory )) {                            //判断是目录
                if ($gain_directory == ".") {
                        $catalog = getcwd ();                        //显示当前目录
                        $catalog = iconv ( "gb2312", "utf-8", $catalog );
                echo "<a href=manage.php?catalog=".urlencode($catalog)."><span class='style1'>锁定</span></a>";
                } elseif ($gain_directory == "..") {
                        $catalog = getcwd () . "\\..";                       //上级目录
                        $catalog = iconv ( "gb2312", "utf-8", $catalog );
                        if($catalog==""){
        echo "<a href=manage.php?catalog=".urlencode($catalog)."><span class='style1'>上级目录</span></a>";
                        }else{
        echo "<a href=manage.php?catalog=".urlencode($catalog)."><span class='style1'>上级目录</span></a>";
                        }
                } else {
                        $catalog = getcwd () . "\\$gain_directory";          //子目录
                        $catalog = iconv ( "gb2312", "utf-8", $catalog );
echo "<a href=manage.php?catalog=".urlencode($catalog).">".iconv( "gb2312", "utf-8", $gain_directory )."</a>";
                }
        } else {
                $ext = substr ( $gain_directory, strrpos ( $gain_directory, "." ) );
                if (strtoupper ( $ext ) == ".html" || strtoupper ( $ext ) == ".gif" || strtoupper ( $ext ) == ".jpg" ) {
                        $catalog = getcwd ();
                        $catalog = iconv ( "gb2312", "utf-8", $catalog );
                        echo   iconv ( "gb2312", "utf-8", $gain_directory ) ;
                } else {
                        echo iconv ( "gb2312", "utf-8", $gain_directory );
                }
        }
```





Note

```
        if (is_dir ( $gain_directory ))
            $file_size = "目录";
        else
        $file_size = filesize ( $gain_directory );
        echo "<td align='center' valign='middle'>$file_size</td>";
        $create_time = date ( "Y-m-d H:i:s", filectime ( $gain_directory ) );
        echo "<td align='center' valign='middle'>" . iconv ( "gb2312", "utf-8", $create_time ) . "</td>";
        $update_time = date ( "Y-m-d H:i:s", filemtime ( $gain_directory ) );
        echo "<td align='center' valign='middle'>$update_time</td>";
        echo "<td align='center' valign='middle'>";
        if ($gain_directory == ".") {
            $catalog = getcwd ();                           //显示当前目录
            echo "删除";
        } elseif ($gain_directory == "..") {
            $catalog = getcwd () . "\\..";                  //上级目录
            echo "删除";
        } else {
            $catalog = getcwd () . "\\$gain_directory";     //子目录
            echo "<a href='delete.php?catalog=".urlencode($catalog)."
    &filename=" . urlencode(getcwd ()) . "' title='删除目录或者文件' >删除</a>";
        }
        echo "</td>";
        echo "</tr>";
    }
    closedir ( $ml );
    ?>
```

（2）创建 delete.php 页，当用户单击“删除”超链接时，在 delete.php 页面接收超链接传递的参数，完成删除操作。其关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
$count=substr_count($_GET['catalog'],'.');
if($count>=1){
    if(unlink($_GET['catalog'])){
        echo "<script>alert('文件删除成功！');
window.location.href='manage.php?catalog=".urlencode($_GET['filename'])."';</script>";
    }else{
        echo "<script>alert('文件删除失败！'); history.back();</script>";
    }
}else{
    if(is_dir($_GET['catalog'])){
        if(@rmdir($_GET['catalog'])){
            echo "<script>alert('目录删除成功！');
window.location.href='manage.php?catalog=".urlencode($_GET['filename'])."';</script>";
        }else{
            echo "<script>alert('目录删除失败！'); history.back();</script>";
        }
    }
}
```



```
?>
```



说明：

在本模块中创建 indexs.php 文件，模拟网站主页。当完成假日公告信息的添加之后，运行 indexs.php 文件，将弹出公告信息，效果如图 7.15 所示。



图 7.15　弹出公告信息效果

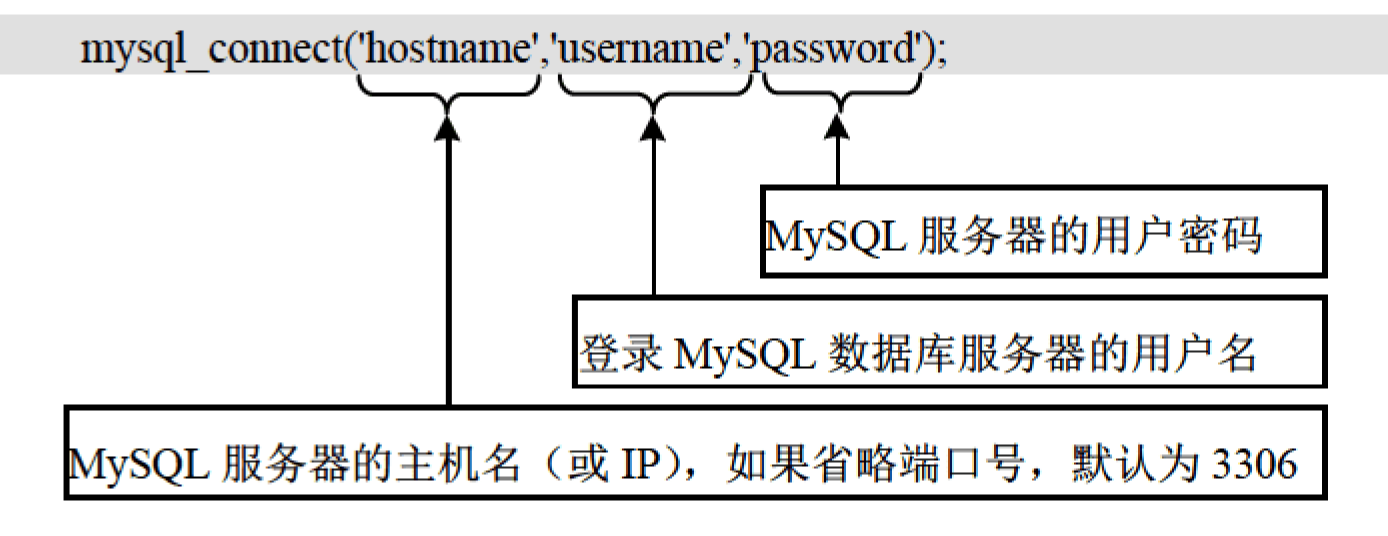
7.7　技 术 提 炼

7.7.1　数据库的连接与数据操作

1. 数据库的连接

（1）要操作 MySQL 数据库，首先必须与 MySQL 服务器建立连接。连接 MySQL 服务器应用的是 mysql_connect()函数，其语法格式如下。

```
mysql_connect('hostname','username','password');
```



该函数的返回值用于表示这个数据库连接。如果连接成功，则函数返回一个资源，为以后执行 SQL 指令做准备。

（2）在连接到 MySQL 数据库服务器之后，接下来使用 mysql_select_db()函数选择数据库。其语法格式如下：



```
mysql_select_db ( string 数据库名[,resource link_identifier] )
```

传入 MySQL 服务器的数据库名称

MySQL 服务器的连接标识

或

```
mysql_query("use 数据库名"[,resource link_identifier]);
```

如果没有指定连接标识符，则使用上一个打开的连接。如果没有打开的连接，本函数将无参数调用 mysql_connect()函数来尝试打开一个数据库并使用。其后的 mysql_query()函数调用都会作用于活动数据库。

例如，在本模块中，与 MySQL 数据库服务器中的 db_jiari 数据库建立连接。conn\conn.php 文件的代码如下：

```
$conn=mysql_connect("127.0.0.1","root","111");                //连接数据库服务器
mysql_select_db("db_jiari",$conn);                            //连接数据库
mysql_query("set names utf8");                                //设置编码
```

2. MySQL 数据操作

对数据库中数据的操作主要是使用 mysql_query()函数，在本模块主要使用 mysql_query()函数实现添加和删除信息的操作。

（1）添加数据

当用户发布假日公告信息时，将向数据库中添加一条假日信息。promulgation_ok.php 文件的关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_POST['name']) and $_POST['Submit']=="发布"){
//添加数据
$insert=mysql_query("insert into tb_01(name,perform) values('".$_POST['name']."','".$_POST
['radio']."')",$conn);
…                                      //省略部分代码
}else{
    echo "<script>alert('请选择上传文件！');window.location.href='promulgation.php';</script>";
}
?>
```

（2）删除数据

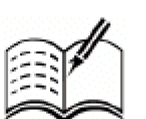
当用户取消假日公告时，将删除数据库中的假日信息。update.php 文件的代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_GET['abrogation'])){
```



```
        $update=mysql_query("delete from tb_01 where id='".$_GET['abrogation']."'",$conn);
        if($update){
            echo "<script>alert('假日信息取消成功！');window.location.href='cancel.php';</script>";
        }else{
            echo "<script>alert('假日信息取消失败！');window.location.href='cancel.php';</script>";
        }
    }
?>
```

Formatted: Line spacing least 12.5 pt



Note

7.7.2　通过超链接传递值

通过超链接传递值主要使用的是<a>标记，其语法格式如下。

```
<a href=URL name=name target=target>链接文字</a>
```

Formatted: Line spacing least 14 pt

a 标记的属性介绍如下。

- ☑ href 属性：指定所链接文件的 URL 路径。这个路径可以是相对路径，也可以是绝对路径。
- ☑ name 属性：指定页面的锚点名称。如果需要链接到对应的锚点位置，需要在锚点名称前添加一个“#”字符。
- ☑ target 属性：指定要打开的链接所使用的浏览器窗口名称。可以使用自定义的窗口名称。也可以使用下面 4 个内置的窗口名称，其中前两个比较常用。
  - ✧ _self：在当前窗口中打开链接文件，此为默认值。
  - ✧ _blank：开启一个新的窗口打开链接文件。
  - ✧ _parent：在父级窗口中打开文件，常用于框架页面。
  - ✧ _top：在顶层窗口中打开文件，常用于框架页面。

本模块中主要在取消公告及删除公告文件中使用了超链接。

（1）假日公告取消页面中的“取消”超链接：

```
<a href="update.php?abrogation=<?php echo $array['id'];?>">取消</a>
```

Formatted: Line spacing least 14 pt

（2）假日公告管理页面中的“删除”超链接：

```
<a href='delete.php?catalog=".urlencode($catalog)."&filename=" . urlencode(getcwd ()) . "' title='删除目录或者文件' >删除</a>
```

Formatted: Line spacing least 14 pt

说明：

获取超链接参数传递的值，应用的是$_GET[]全局变量。

7.7.3　单文件上传

本模块中上传的单文件是公告信息文件，主要使用了 mkdir() 函数和 move_uploaded_file()函数。下面分别对其进行介绍。





Note

（1）mkdir()函数

判断某文件是否存在，并且是否可写，如果满足上述条件则返回 True，否则返回 False。其语法格式如下：

```
bool mkdir(string pathname [, int mode])
```

其中参数说明如表 7.1 所示。

表 7.1　mkdir()函数的参数说明

参　　数	说　　明
pathname	必要参数。用于指定新建目录的名称
mode	可选参数。用于指定新建目录的模式

Formatted: 表栏名

（2）move_uploaded_file()函数

该函数应用 POST 方法实现文件的上传。其语法格式如下：

```
bool move_uploaded_file(string filename, string destination)
```

其中参数说明如表 7.2 所示。

表 7.2　move_uploaded_file()函数的参数说明

参　　数	说　　明
filename	必要参数。指定要上传的文件地址
destination	必要参数。上传到服务器后的存储目录及名称

注意：

应用 POST 方法上传文件时，必须在上传表单的<form>标记中添加 enctype="multipart/form-data"属性。

7.7.4　多图片上传

本模块在上传公告图片时，采用了多图片上传方式，使用 for 循环语句循环获取图片信息，然后使用 move_upload_file()函数执行上传操作。

for 语句是 PHP 中最复杂的循环控制语句，拥有 3 个条件表达式。其语法格式如下：

```
for (expr1; expr2; expr3){
      statement
}
```

其中参数说明如表 7.3 所示。

表 7.3　for 循环语句中的参数介绍

参　　数	说　　明
expr1	必要参数。第一个条件表达式，在第一次循环开始时被执行
expr2	必要参数。第二个条件表达式，在每次循环开始时被执行，决定循环是否继续
expr3	必要参数。第三个条件表达式，在每次循环结束时被执行
statement	必要参数。满足条件后，循环执行的语句

Formatted: Line spacing least 13 pt

Formatted: Line spacing least 13 pt

Formatted: Line spacing least 13 pt

Formatted: Line spacing least 13 pt

Formatted: Line spacing least 13 pt





其执行的过程：首先执行表达式 1；然后执行表达式 2，并对表达式 2 的值进行判断，如果值为真，则执行 for 循环语句中指定的内嵌语句，如果值为假，则结束循环，跳出 for 循环语句；最后执行表达式 3（切记是在表达式 2 的值为真时），返回表达式 2 继续循环执行。for 循环语句的操作流程如图 7.16 所示。

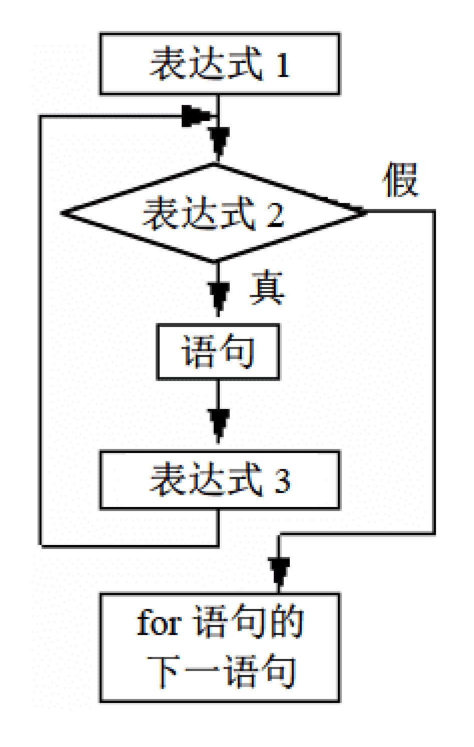


图 7.16　for 循环语句的流程图

7.7.5　读取指定目录中的目录及文件

读取指定目录中的目录及文件主要使用 opendir()函数、getcwd()函数、iconv()函数以及 while 循环语句，循环输出目录文件。

（1）opendir()函数

打开指定的目录，并返回该目录的标识。其语法格式如下：

```
resource opendir(string path)
```

参数 path 为该函数所要打开目录的路径。

（2）getcwd()函数

返回当前的工作目录。其语法格式如下：

```
string getcwd(void)
```

该函数的参数为空。

（3）iconv()函数

将一种已知的字符编码格式 A 转换成另外一种编码格式 B。其语法格式如下：

```
string iconv ( string in_charset, string out_charset, string str )
```

参数说明如下。

in_charset：一种已知将被转换的编码格式 A。

out_charset：另一种将要转换成的编码格式 B。

str：指定被转换的字符串。该函数转换成功返回字符串，否则返回 false。

（4）while 循环语句

while 循环语句的作用是反复地执行某一项操作，是循环控制语句中最简单的一种，





也是最常用的一种。while 循环语句对表达式的值进行判断，当表达式的值为非零时，执行 while 语句中的内嵌语句；当表达式的值为 0 时，则不执行 while 语句中的内嵌语句。该语句的特点是：先判断表达式，后执行语句。while 循环控制语句的操作流程如图 7.17 所示。

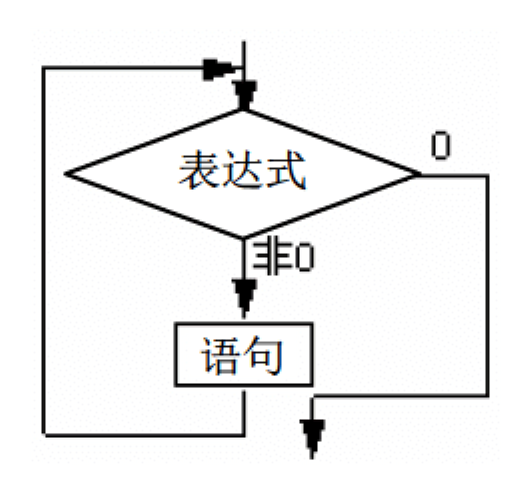


图 7.17 while 循环控制语句的操作流程

其语法格式如下：

```
while (expr){                    /*
    statement;                   先判断条件，当条件满足时执行语句块，否则
                                 不向下执行
}                                */
```

只要 while 表达式 expr 的值为 TRUE，就重复执行嵌套中的 statement 语句；如果 while 表达式的值一开始就是 FALSE，则循环语句一次也不执行。

7.7.6 jQuery 控制弹出公告信息

在假日公告管理系统中，控制公告信息的输出应用的是 jQuery 技术。下面对具体应用的方法进行介绍。

（1）$()：用于替代 document.getElementById()。其语法格式如下：

```
var someElement = $("#myId");
```

例如：通过 jQuery 获取 DOM 对象中的元素，代码如下：

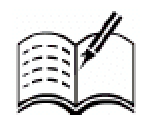
```
01    $("div m");
02    $("div.admin");
03    $("div #mysqlid ");
04    $("table a",content);
```

第一行代码获取所有标签下的 m 元素；第二行代码获取 class 为 admin 的元素；第三行代码获取标签下面 id 为 mysqlid 的元素；第四行代码获取 content 为上下文的 table 中所有的连接元素。

（2）slideToggle(speed,[callback])：通过高度变化来切换所有匹配元素的可见性，并在切换完成后可选地触发一个回调函数。

这个动画效果只调整元素的高度，可以使匹配的元素以“滑动”的方式隐藏或显示。参数说明如下。





- ☑ speed（String,Number）：3 种预定速度之一的字符串（"slow", "normal", or "fast"）或表示动画时长的毫秒数值（如 1000）
- ☑ callback (Function)：可选参数，在动画完成时执行的函数。

例如，用 600 毫秒缓慢地将段落滑上或滑下。

```
$("p").slideToggle("slow");
```

又如，用 200 毫秒快速将段落滑上或滑下，之后弹出一个对话框。

```
$("p").slideToggle("fast",function(){
    alert("Animation Done.");
});
```

（3）ready(fn)：当 DOM 载入就绪，可以查询及操纵时，绑定一个要执行的函数。这个方法是 window.load 事件的替代方法。其语法格式如下：

```
ready(fn)
```

参数 fn (Function)为在 DOM 就绪时执行的函数。可以为这个参数任意起一个名字，并因此可以不再担心命名冲突而放心地使用$别名。

说明：

在使用 ready(fn)时，必须确保在<body>元素的 onload 事件中没有注册函数，否则不会触发$(document).ready()事件。可以在同一个页面中无限次地使用$(document).ready()事件。其中注册的函数会按照代码中的先后顺序依次执行。

例如，通过$(document).ready(fn)代替 body 标签的 onload 事件，代码如下：

```
$(document).ready(function(){
        alert("jQuery");
});
```

使用$(document).ready()的简写，同时内部的 jQuery 代码依然使用$作为别名，而不必管全局的$是什么。其 jQuery 代码如下：

```
jQuery(function($) {
  //可以在这里继续使用$作为别名...
});
```

上述$(document).ready()的简写替换的是如下代码：

```
<body onload="alert('jQuery');">
```

（4）css(name)：访问第一个匹配元素的样式属性。返回值为 string，参数 name 为要访问的属性名称。

例如，取得第一个段落的 color 样式属性的值。

```
$("p").css("color");
```

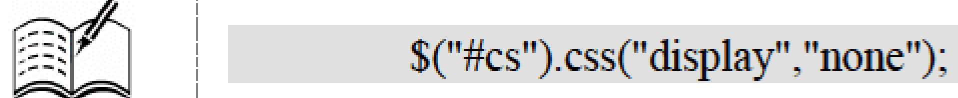
（5）css(name,value)：在所有匹配的元素中，设置一个样式属性的值。数字将自动转



化为像素值。

其中，参数 name 为属性名，参数 value (String, Number)是属性值。

例如，初始状态下隐藏下拉列表。



```
$("#cs").css("display","none");        //初始状态下使城市下拉列表不可见
```

Note

在假日公告管理系统中，需要在输出公告信息的网页中载入 jQuery 库文件，编写 JavaScript 脚本，控制公告信息的输出。其关键代码如下：

```
<script src="js/jquery.js"></script>
<script>
        function showLayerNotice(){
                $("#Layer110").slideToggle(1000);
        }
        $(document).ready(function(){
                $("#Layer110").css("left", (screen.width-520)/2);
                setTimeout("showLayerNotice()", 1300) ;
         });
</script>
```

同时还要创建 div 标签，输出公告信息。在输出公告信息之前，要读取数据库中的数据，判断是否输出公告信息。如果指定字段值为 1，则包含上传的 html 文件，即输出公告信息；否则不输出 div 标签中的内容。其关键代码如下：

```
<?php
include("conn/conn.php");                                              //连接数据库
$result=mysql_query("select perform from tb_01",$conn);                //执行查询语句
$myrow=mysql_fetch_array($result);                                     //获取查询结果
if($myrow['perform']==1){//判断指定字段的值，如果等于 1 则输出公告信息，否则不输出
?>
<div id="Layer110"
style="position: absolute; border:3px solid #CCCCCC; left: 50px; top: 50px; z-index: 1; font-size: 0px; display: none">
<?php
        include("upfile/index.html");                                  //包含公告文件
?>
</div>
<?php
}
?>
```

第 8 章

投票系统模块

（自学视频、源程序：配套资源\mr\8\）

针对某一主题，在进行网上调查时，投票系统必不可少。通过该系统可以较为客观、真实地收集、分析广大用户对该主题的意见及想法，以便作出正确的判断，从而有针对性地加以改进或完善，为用户提供更好的服务。在本章中将向广大读者介绍一些投票的实现方法和分析投票结果的方法，以及在投票中如何限制重复投票等技术。通过本章的学习，读者能够学到以下内容。

- 单选投票与多选投票的关系
- 通过柱形图与饼图分析投票结果
- 通过 IP 限制重复投票
- 利用 Cookie 变量限制重复投票
- 动态生成投票主题及选项



8.1 投票系统概述



Note

8.1.1 模块概述

投票系统的主要作用是根据一个主题，以提供的投票选项为条件，选出正确、合理的一项或者多项内容，从而更好地了解广大用户对该主题的意见及想法。为了使其功能更加完善、合理，本系统还开发了后台管理功能，对投票和网站进行管理。

8.1.2 功能结构

投票系统的功能结构如图 8.1 所示。

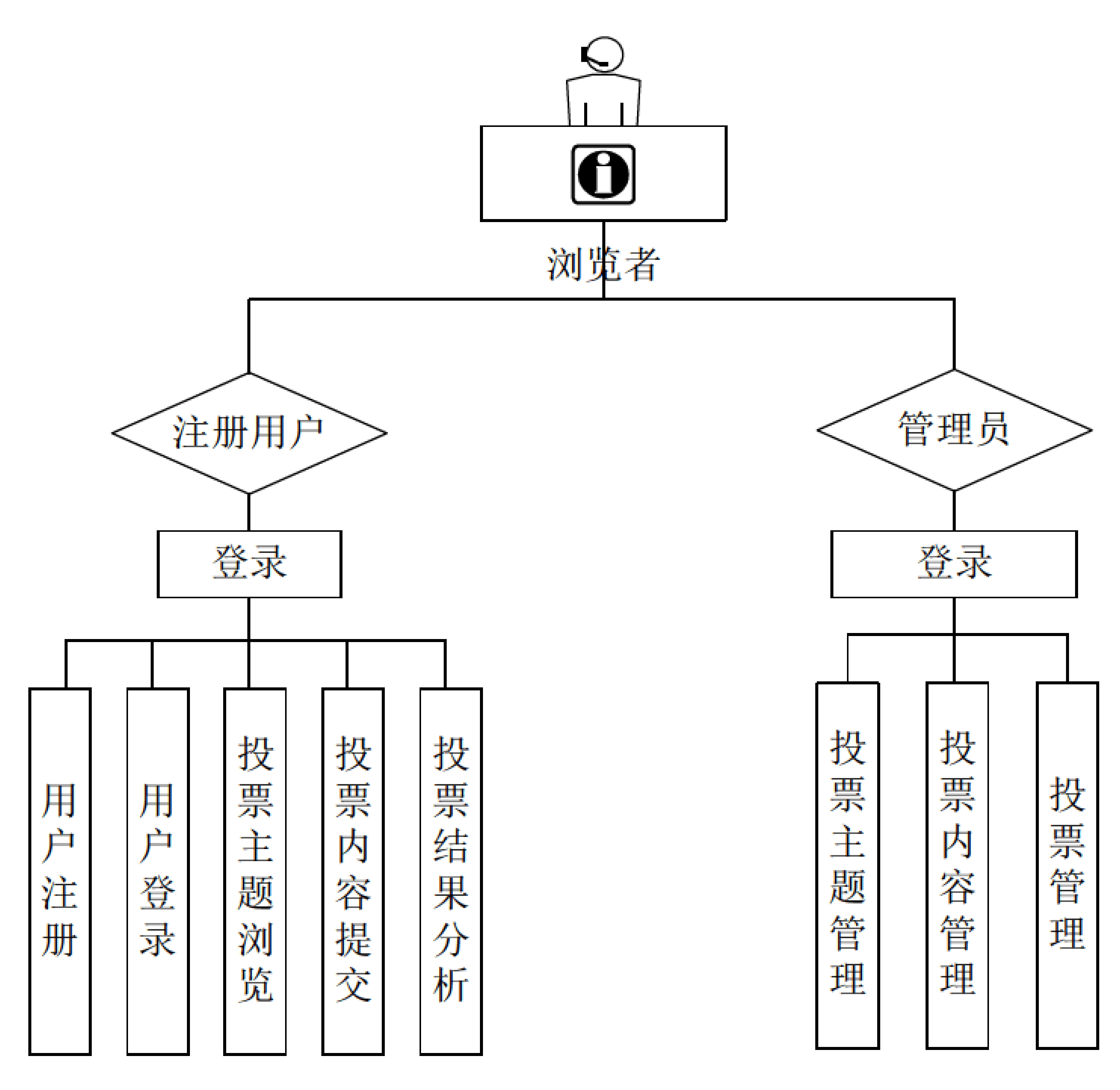


图 8.1　投票系统的功能结构图

8.1.3 程序预览

投票系统是由多个功能模块组成的，由于篇幅有限，下面仅列出几个典型功能模块的运行效果图，让读者对本系统有一个初步的了解和认识。

投票主题浏览页面运行效果如图 8.2 所示，其主要功能是给浏览者提供一个投票的平台。







图 8.2 投票主题浏览页面运行效果

通过柱形图分析投票结果，其运行效果如图 8.3 所示。

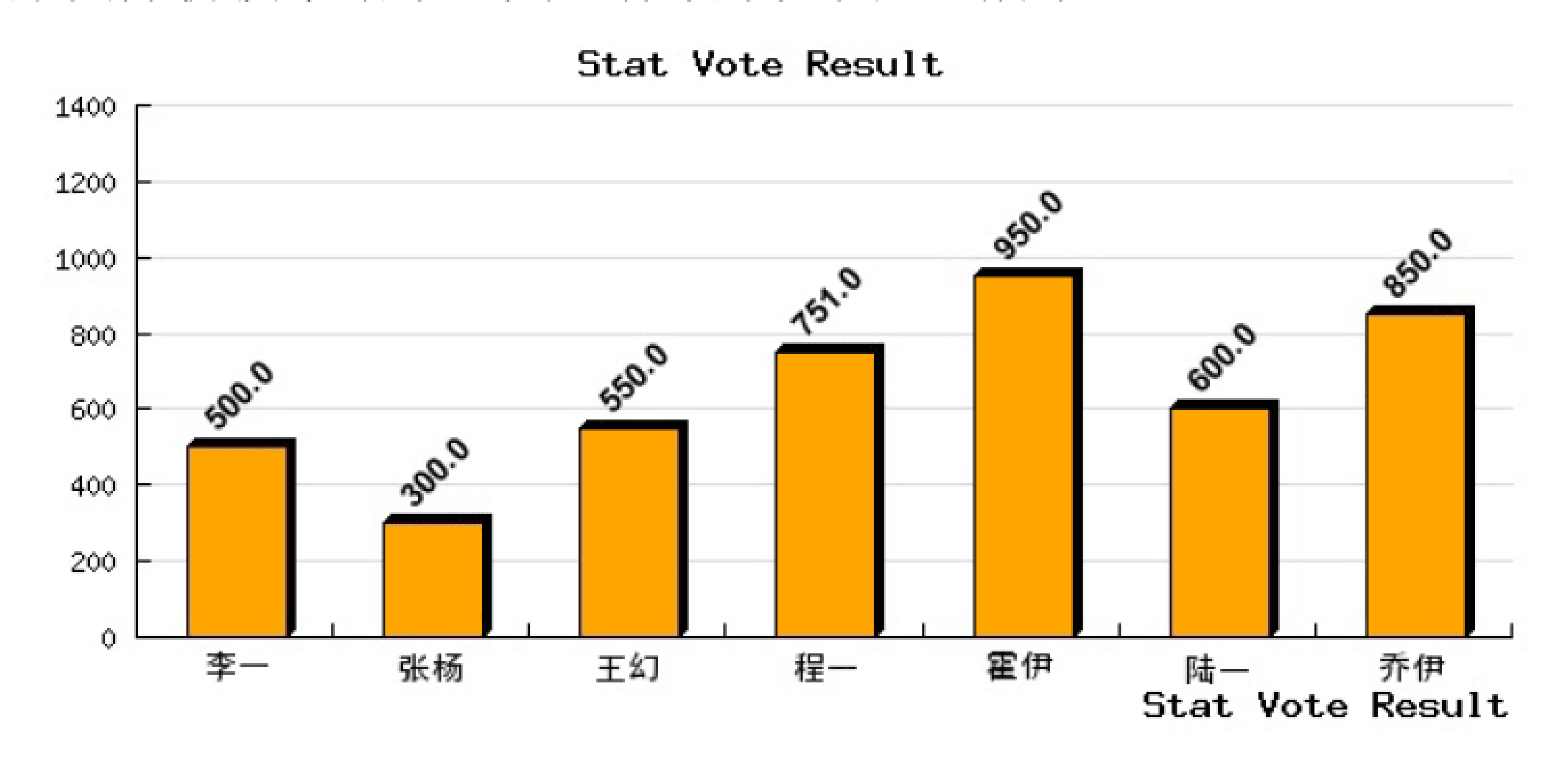


图 8.3 通过柱形图分析投票结果

通过 3D 饼图分析投票结果，其运行效果如图 8.4 所示。

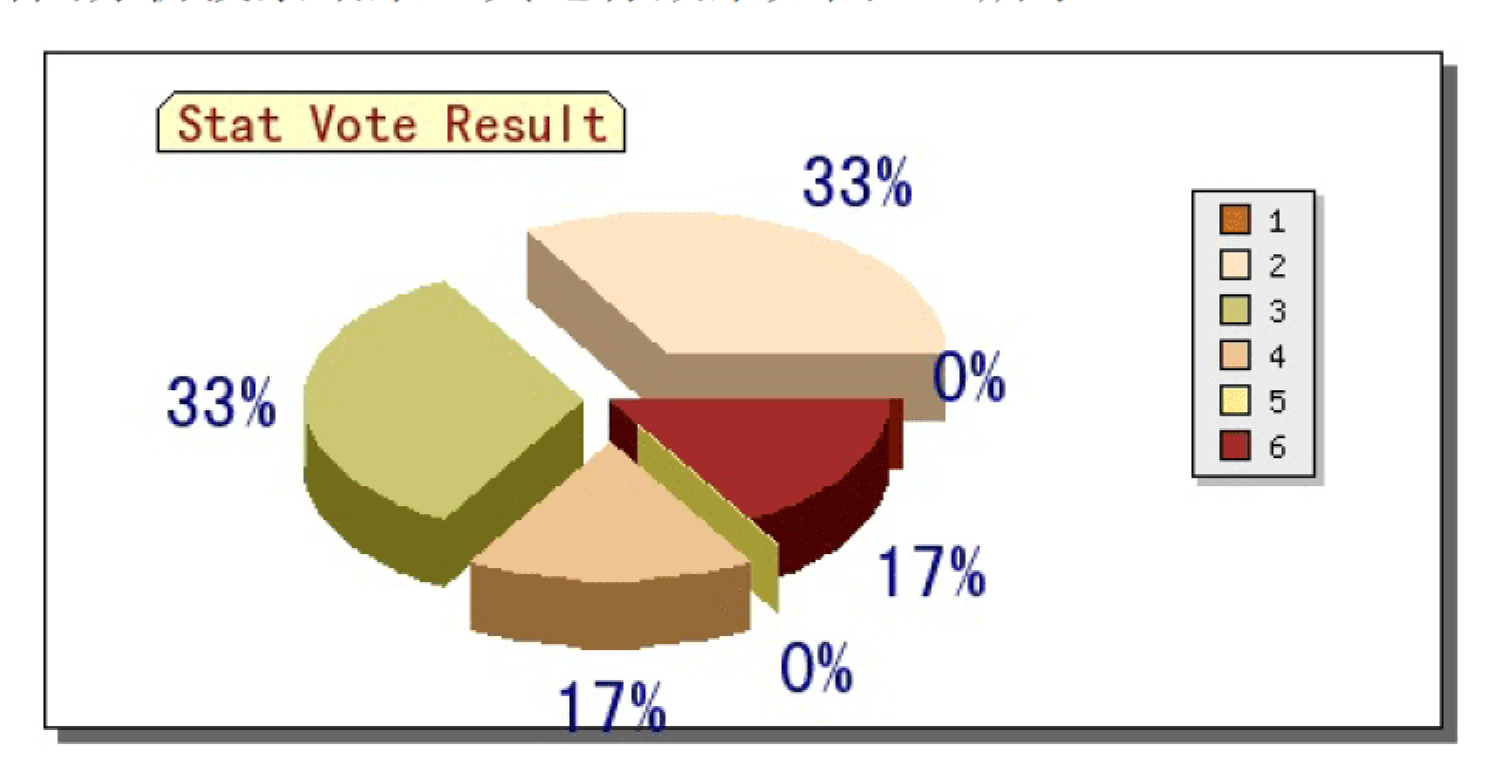


图 8.4 通过 3D 饼图分析投票结果

后台管理首页的运行效果如图 8.5 所示，其主要功能是对投票的主题内容和投票进行管理。



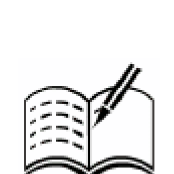




图 8.5　后台管理首页运行效果

管理投票主题页面的运行效果如图 8.6 所示，其主要功能是给浏览者提供投票的主题，以及对已有投票主题的删除。



图 8.6　管理投票主题页面运行效果

8.2　数据库设计

8.2.1　数据库设计

投票系统模块使用的数据库是 db_vote，其中包含 6 个数据表，分别是 tb_voter（用户信息表）、tb_vote_more（多选投票表）、tb_vote_more_ip（参与多选投票者的 ip 表）、tb_vote_odd（单选投票表）、tb_vote_odd_ip（参与单选投票者的 ip 表）、tb_vote_subject（投票主题表）。db_vote 数据库结构如图 8.7 所示。



服务器: localhost ▸ 数据库: db_vote					
表	操作	记录数	类型	整理	说明
tb_voter		5	MyISAM	gb2312_chinese_ci	存储注册的用户信息
tb_vote_more		7	MyISAM	gb2312_chinese_ci	存储多选投票的内容
tb_vote_more_ip		0	MyISAM	gb2312_chinese_ci	通过IP控制多选投票中的重复投票
tb_vote_odd		13	MyISAM	gb2312_chinese_ci	存储单选投票内容
tb_vote_odd_ip		5	MyISAM	gb2312_chinese_ci	通过IP控制单选投票中的重复投票
tb_vote_subject		6	MyISAM	gb2312_chinese_ci	存储添加的投票主题
6个表	总计	48	MyISAM	gb2312_chinese_ci	

图 8.7　投票系统数据库结构



8.2.2　数据表设计

下面对投票系统使用的 db_vote 数据库中的数据表结构进行分析。由于篇幅有限，在此仅以 tb_vote_more 和 tb_vote_subjet 为例进行介绍。

tb_vote_more（多选投票表）的结构如图 8.8 所示。

服务器: localhost ▸ 数据库: db_vote ▸ 表 : tb_vote_more							
字段	类型	整理	属性	Null	默认	额外	说明
tb_vote_id	int(10)			否		auto_increment	自动编号
tb_vote_subject	varchar(80)	gb2312_chinese_ci		否			投票主题
tb_vote_content	mediumtext	gb2312_chinese_ci		否			投票内容
tb_vote_counts	varchar(50)	gb2312_chinese_ci		否			投票数
tb_vote_type	varchar(50)	gb2312_chinese_ci		否			投票方式
tb_vote_explain	mediumtext	gb2312_chinese_ci		否			投票说明
tb_vote_photo	varchar(80)	gb2312_chinese_ci		否			投票内容的图片

图 8.8　多选投票表

tb_vote_subject（投票主题表）的结构如图 8.9 所示。

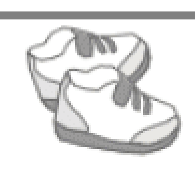
服务器: localhost ▸ 数据库: db_vote ▸ 表 : tb_vote_subject							
字段	类型	整理	属性	Null	默认	额外	说明
tb_vote_subject_id	int(10)			否		auto_increment	投票主题的ID
tb_vote_subject_name	varchar(80)	gb2312_chinese_ci		否			投票主题名称
tb_vote_subject_text	text	gb2312_chinese_ci		否			投票主题描述

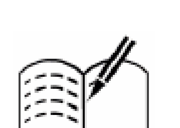
图 8.9　投票主题表

8.2.3　连接数据库

由于本模块大部分页面都需要使用数据库，如果每页都编写相同的数据库连接代码，会显得十分繁琐，所以在此将数据库连接代码单独存入一个 PHP 文件 conn.php 中，并将该文件存储于 conn 文件夹中，在需要与数据库连接的页面中，使用包含语句调用该文件即可。该模块实现与数据库连接的代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or die('连接失败:' . mysql_error());    // 连 接
服务器
mysql_select_db("db_vote",$conn) or die('数据库选择失败:' . mysql_error());    //选择数据库
mysql_query("set names gb2312");                                    //设置数据库编码格式
?>
```





Note

8.3 投　　票

在本模块中，浏览者想要对某一主题进行投票，必须先进行注册。注册成功后，以指定的用户身份登录系统，然后才可以进行投票。在本节中将对用户的注册、登录、投票主题浏览和投票进行详细讲解。

8.3.1 用户注册

用户注册的功能是存储投票者的个人信息，同样也是为投票者提供一个登录本系统的钥匙。用户注册模块的运行效果如图 8.10 所示。



图 8.10　用户注册页面

首先，创建一个 register.php 文件，在该文件中添加用户的个人信息，并且应用 JavaScript 脚本对添加的信息进行判断，将数据提交到 register_ok.php 文件中。其中应用的 JavaScript 脚本语句如下：

```
<script>
function check_email(tb_voter_mail){
	var str=tb_voter_mail;
	var Expression=/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;	//判断邮箱地址的格式是否正确
	var objExp=new RegExp(Expression);
	if(objExp.test(str)==true){
		return true;
	}else{
		return false;
	}
}
</script>
<script language="JavaScript" type="text/javascript">
```



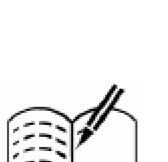

```
function check_input(form){
        if(form.tb_voter_name.value==""){
                alert("请填写会员名称！");
                form.tb_voter_name.select();
                return(false);
        }
        if(form.tb_voter_pass.value==""){
                alert("请输入密码！");
                form.tb_voter_pass.focus();
                return(false);
        }
        if(form.tb_voter_mail.value==""){
               alert("请输入 E-mail 地址!");
               form.tb_voter_mail.select();
               return(false);
        }
        if(!check_email(form.tb_voter_mail.value)){
               alert("您输入的 E-mail 地址的格式不正确!");
               form.tb_voter_mail.focus();
               return(false);
        }
        if(form.tb_voter_tel.value==""){
                alert("请输入电话！");
                form.tb_voter_tel.select();
                return(false);
        }
        if(form.tb_voter_address.value==""){
                alert("请填写地址！");
                form.tb_voter_address.select();
                return(false);
        }

        return(true);
}
</script>
```

然后，创建 register_ok.php 文件，将表单提交的数据存储到指定的数据表中。其代码如下：

```
<?php
session_start();
include_once("conn/conn.php");
if (isset($_POST['submit'])){                              //处理注册用户提交的数据
        $tb_voter_name=trim($_POST['tb_voter_name']);
        $tb_voter_pass=md5($_POST['tb_voter_pass']);
        $tb_voter_mail=trim($_POST['tb_voter_mail']);
        $tb_voter_tel=trim($_POST['tb_voter_tel']);
        $tb_voter_address=trim($_POST['tb_voter_address']);
        $tb_voter_ip=getenv("REMOTE_ADDR");            //获取投票者的 IP
$query=mysql_query("insert into tb_voter(tb_voter_name,tb_voter_pass,tb_voter_mail,tb_voter_ tel,tb_voter_
```





Note

```
address,tb_voter_ip)
        values('$tb_voter_name','$tb_voter_pass','$tb_voter_mail','$tb_voter_tel','$tb_voter_address','$tb_voter_ip')",$conn);
            if($query){
                $_SESSION['tb_voter_name']=$tb_voter_name;     //通过 seesion 变量存储投票用户的姓名
                $_SESSION['tb_voter_mail']=$tb_voter_mail;     //通过 seesion 变量存储投票用户的邮箱
                echo "<script>alert('注册成功!');window.location.href='main.php';</script>";
            }else{
                echo "<script language='javascript'>alert('对不起,注册失败!');history.back();</script>";
            }
        }
    ?>
```

8.3.2 用户登录

用户登录模块主要是供投票系统的注册用户进行登录操作。该页面主要由两个文件组成，一个是用户登录文件，包括用户名、密码；另一个是用户登录的处理文件。用户登录页面如图 8.11 所示。

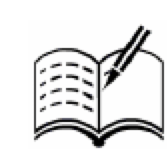


图 8.11 用户登录页面

在用户登录文件中，创建一个表单，其中包括用户名和密码两个文本框，将表单中的内容提交到 enter_ok.php 文件中。在 index.php 文件中，创建表单和表单元素的关键代码如下：

```
<form action="enter_ok.php" method="post" name="form1" id="form1" onSubmit="return check_user(this)">
        <tr>
        <td width="87" align="center">用户名：</td>
        <td width="139"><input type="text" name="tb_user" size="18"   /></td>
        <td width="69" align="center">密码：</td>
          <td width="148"><input type="password" name="tb_pass" size="18" /></td>
          <td width="63"><input type="submit" name="Submit" value="登录"></td>
        <td width="186">今天是：<?php echo date("Y-m-d");?></td>
        <td width="58"> </td>
```






```
        </tr>
    </form>
```

在 enter_ok.php 文件中，实现对表单提交的数据进行处理。主要是判断用户提交的用户名和密码是否正确，如果正确则可以进行登录，否则将不能登录。代码如下：

```
<?php
session_start();
include("conn/conn.php");                   //连接数据库
$tb_user=$_POST['tb_user'];                 //获取提交用户名
$tb_pass=md5($_POST['tb_pass']);            //获取提交密码
if(isset($_POST['tb_user']) && isset($_POST['tb_pass'])){
    $query=mysql_query("select * from tb_voter where tb_voter_name='$tb_user' and tb_voter_pass='$tb_pass'");
    if($query){
        $_SESSION["tb_user"]=$tb_user;
        echo "<script>alert('登录成功！');window.location.href='main.php';</script>";
    }else{
        echo "<script>alert('您输入的用户名或密码不正确！');window.location.href='index.php';</script>";
    }
}else{
    echo "<script>alert('请您正确登录，谢谢!');window.location.href='index.php';</script>";
}
?>
```

8.3.3　投票主题浏览

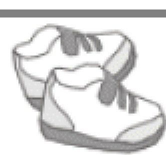
在投票主题浏览页面中，将输出本系统提供的投票主题，同时还将显示一些超链接，用于进入相应的投票内容提交页面中。投票主题浏览页面的运行效果如图 8.12 所示。



图 8.12　投票主题浏览页面

投票主题浏览是通过 main.php 文件来完成。首先，判断用户是否正确登录，如果正确登录，则可以进入本系统中，否则将弹出提示信息“请您正确登录!”，返回到用户登录页面。

然后，应用 SELECT 查询语句，从投票主题表中读取出所有投票主题的数据，将数据进行循环输出，并且设置超链接，链接到投票内容提交页面。main.php 的关键代码如下：





Note

```
<?php session_start();
include("conn/conn.php");
if(isset($_SESSION['tb_user'])) {
?>
…
<?php
        $result=mysql_query("select * from tb_vote_subject");
        while($myrow=mysql_fetch_array($result)){        //循环输出投票类别
?>
   <tr bgcolor="#ECF6FF">
        <td width="39"> </td>
        <td colspan="2" class="style1">
<a href="vote.php?vote_subject=<?php echo $myrow['tb_vote_subject_name'];?>">
<?php echo $myrow['tb_vote_subject_name'];?> </a></td>
        <td width="523" height="50"><span class="style2"> 
说明:  <?php echo $myrow['tb_vote_subject_text'];?> </span></td>
   </tr>
   <?php }?>
…
<?php }else{
        echo "<script>alert('请您正确登录！');window.location.href='index.php';</script>";
}?>
```

投票主题浏览功能创建完成后，接下来就是创建投票内容提交页面。本系统投票内容的创建是在后台完成的。

8.3.4 投票内容提交

投票内容提交模块的功能是：根据超链接栏目标识中的变量值，从数据库中读取出对应主题的数据，将投票的内容进行输出；然后创建表单，实现投票的提交；最后输出投票的结果，并且设置“投票结果”超链接，用于对投票结果进行分析。投票内容提交页面的运行效果如图 8.13 所示。



图 8.13　投票内容提交页面





下面详细介绍投票内容提交的实现过程。在 vote.php 文件中，整体上分为如下 4 个部分。

（1）判断用户是否正常登录，如果是则可以进行登录，否则将弹出提示信息“请您正确登录！”，并返回到用户登录页面。其关键代码如下：

```
<?php
session_start();                          //初始化 SESSION 变量
include("conn/conn.php");                 //包含数据库连接文件
if(isset($_SESSION['tb_user'])){          //判断用户是否正常登录
?>
…
<?php }else{
    echo "<script>alert('请您正确登录！');window.location.href='index.php';</script>";
}?>
```

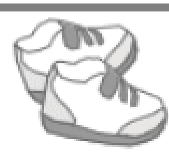
（2）根据超链接中传递的变量值，输出对应的投票主题和说明的内容。代码如下：

```
<?php
$result=mysql_query("select * from tb_vote_subject where tb_vote_subject_name='".$_GET['vote_subject']."'");
while($myrow=mysql_fetch_array($result)){                    //循环输出查询结果
$tpzt=$myrow['tb_vote_subject_name'];
?>
<tr bgcolor="#ECF6FF">
    <td colspan="2" class="style1">
        <a href="main.php"><?php echo $myrow['tb_vote_subject_name'];?></a>
    </td>
     <td width="529" height="50"><span class="style1"> 说明: 
        <?php echo $myrow['tb_vote_subject_text'];?> </span>
    </td>
</tr>
<?php
}
?>
```

（3）输出投票选项中属于单选类的投票内容。在单选类的投票中，又进一步分为图片类投票和无图片类投票两类。

首先，判断投票选项是否属于单选类别，如果属于单选类则继续执行，创建一个 form 表单，用于提交投票选项，将结果提交到 tb_vote_ok.php 文件中。代码如下：

```
<?php
$query1="select * from tb_vote_odd where tb_vote_subject='".$_GET['vote_subject']."'";
$result1=mysql_query($query1);                  //执行查询语句
$myrow1=mysql_fetch_array($result1);            //获取查询结果
$tb_vote_type=$myrow1['tb_vote_type'];          //获取类型
$tb_vote_photo=$myrow1['tb_vote_photo'];        //获取图片
if($tb_vote_type=="单选"){
?>
<table cellpadding="0" cellspacing="0" bordercolor="#FFFFFF" background="images/bg3.gif" bgcolor="#FFFFFF">
<form name="form1" method="post" action="tb_vote_ok.php" enctype="multipart/form-data">
```



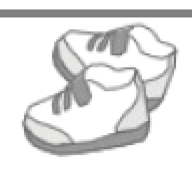


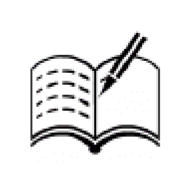
Note

```
		<tr>
			<td height="25" align="center">单选区</td>
			<td height="25" align="center"> 投票主题(<?php echo $_GET['vote_subject'];?>)</td>
<td align="center" ><a href="see_vote_result.php?vote_subject=<?php echo $_GET['vote_subject'];?>">投票结果</a>
			</td>
		</tr>
```

然后，判断 tb_vote_photo 字段的值是否为空，如果为空则说明是无图片投票，则执行下面的操作。应用 while 循环语句，输出投票选项，并且设置单选按钮来提交投票。关键代码如下：

```
	<?php
	if ($tb_vote_photo==""){				//判断 tb_vote_photo 的值是否为空
		$query2="select * from tb_vote_odd where tb_vote_subject='".$_GET['vote_subject']."'";
		$result2=mysql_query($query2);		//循环输出数据
		while($myrow2=mysql_fetch_array($result2)){
	?>
		<tr>
			<td width="163" align="center"><span class="style2">  
				<?php echo $myrow2['tb_vote_content'];?></span></td>
			<td width="194">
				<input name="tb_vote_id" type="radio" value="<?php echo $myrow2['tb_vote_id'];?>">
		<input type="hidden" name="tb_vote_subject" value="<?php echo $myrow2['tb_vote_subject'];?>">
			</td>
			<td align="center">
				<table width="224" cellpadding="0" cellspacing="0">
					<tr>
				<td width="161" align="center" class="style2"><?php echo $myrow2['tb_vote_content'];?></td>
						<td width="61" class="style2"> <?php echo $myrow2['tb_vote_counts'];?></td>
					</tr>
				</table>
			</td>
		</tr>
	<?php
		}
	?>
		<tr>
			<td> </td>
			<td><input type="submit" name="Submit" value="提交"></td>
			<td> </td>
		</tr>
	  </form>
	<?php
	}else{
	?>
```

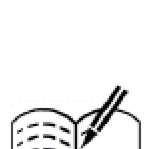




如果 tb_vote_photo 字段的值不为空，则说明是图片投票，执行下面的操作。应用 while 循环语句，从数据库中循环读取投票选项的数据，并且对图片进行排列输出。这里又创建了一个 form 表单，用于提交投票选项，同样将投票结果提交到 tb_vote_ok.php 文件中。关键代码如下：

```
<?php
$query2="select * from tb_vote_odd where tb_vote_subject='".$_GET['vote_subject']."'";
$result2=mysql_query($query2);
$m=0;               //定义变量，用于图片的排列输出
while($myrow2=mysql_fetch_array($result2)){
        $m++;       //用于图片的排列输出
?>
<table width="150" border="0" align="left" cellpadding="0" cellspacing="0">
        <tr>
                <td height="25" align="center">
<table border="1" cellpadding="0" cellspacing="0" bordercolor="#FFFFFF" bgcolor="#FFFFFF">
<form name="form2" method="post" action="tb_vote_ok.php" enctype= "multipart/form-data">
        <tr>
                <td colspan="2" align="center" valign="bottom" bgcolor="#F3F3F3">
                <img src="<?php echo substr($myrow2['tb_vote_photo'],3,200);?>" width="100" height="80" ></td>
        </tr>
        <tr>
                <td height="20" colspan="2" align="center" bgcolor="#F3F3F3">
                        <input name="tb_vote_id" type="hidden" value="<?php echo $myrow2['tb_vote_id'];?>">
                        <?php echo $myrow2['tb_vote_content'];?>
          <input type="hidden" name="tb_vote_subject" value="<?php echo $myrow2['tb_vote_subject'];?>">
                        <input type="hidden" name="Submit" value="提交">
                </td>
        </tr>
        <tr>
                <td height="30" align="center" bgcolor="#F3F3F3">
                        <input name="imageField" type="image" id="imageField" src="images/1.gif"></td>
                <td height="24" align="center" bgcolor="#F3F3F3"><a href="#">
                        <img src="images/2.gif" width="52" height="18" border="0"
onClick="MM_openBrWindow('see_voter_information.php?odd_id=<?php echo $myrow2['tb_vote_id'];?>','
查看投票资料','width=300,height=300')"></a>
                </td>
        </tr>
        <tr>
                <td height="20" colspan="2" align="center" bgcolor="#F3F3F3">
                        <?php echo $myrow2['tb_vote_counts'];?>
                </td>
        </tr>
    </form>
```






```
</table>
                </td>
        </tr>
</table>
<?php
if ($m%5==0)    //当变量$m%5 的值等于 0 时，执行下面的内容
        echo "<br /><br /><br /><br /><br /><br /><br /><br /><br />";
}
?>
                                </td>
                                <td width="1%"> </td>
                        </tr>
                </table>
                </td>
        </tr>
<?php   }  ?>
</table>
<?php
}
?>
```

到此，第（3）部分输出单选类投票内容介绍完毕。

（4）输出多选类投票的内容，应用的方法与第（3）部分相同，这里不再赘述。

无论是单选或者多选的投票内容，都将提交到 tb_vote_ok.php 文件中，将投票结果添加到指定的数据表中，并通过客户端的 IP 地址对重复投票进行限制。在 tb_vote_ok.php 中完成投票内容的提交可以分成如下几个步骤：

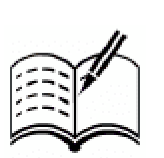
（1）初始化 Session 变量，连接数据库，获取系统的当前时间，获取客户端的 IP 地址。

```
<?php
session_start();
include("conn/conn.php");
$data=date("Y-m-d h:i:s");
$ip=getenv("REMOTE_ADDR");        //获取 IP
```

（2）执行单选投票内容提交的操作，判断是否有数据提交，通过 IP 判断是否是重复投票。代码如下：

```
if (isset($_POST['Submit'])){
        $tb_vote_id=$_POST['tb_vote_id'];
        if ($_POST['tb_vote_id']==""){
                echo "<script>alert('您没有选择投票的内容!');
window.location.href='vote.php?vote_subject=".$_POST['tb_vote_subject']."';</script>";
        }else{
$query="select * from tb_vote_odd_ip where tb_vote_subject='".$_POST['tb_vote_subject']."' and
tb_vote_ip='$ip'";
                $result=mysql_query($query);
                $row=mysql_num_rows($result);
                if ($row>0){            //判断此 IP 是否重复
                        echo "<script>alert('".$ip." 您不可以重复投票!');
```





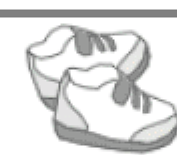

```
window.location.href='vote.php?vote_subject=".$_POST['tb_vote_subject']."';</script>";
        }else{
```

（3）执行单选投票的数据更新操作，并将客户端的 IP 地址存储到指定的数据表中。代码如下：

```
$querys="update tb_vote_odd set tb_vote_counts=tb_vote_counts+1 where tb_vote_id='$tb_vote_id'";
$result=mysql_query($querys);
$query=mysql_query("select * from tb_vote_odd where tb_vote_id='$tb_vote_id'");
$myrow=mysql_fetch_array($query);
$tb_vote_subject=$myrow['tb_vote_subject'];
$tb_vote_content=$myrow['tb_vote_content'];
$querys="insert into tb_vote_odd_ip(tb_vote_subject,tb_vote_content,tb_vote_ip,tb_vote_date,tb_vote_id )
        values('$tb_vote_subject','$tb_vote_content','$ip','$data','$tb_vote_id')";
 $result=mysql_query($querys);
 echo "<script>alert('投票提交成功!');
     window.location.href='vote.php?vote_subject=".$_POST['tb_vote_subject']."';</script>";
        }
    }
}
```

（4）执行多选投票内容的提交，并通过 Cookie 来限制重复投票。代码如下：

```
if (isset($_POST['Submit2'])){
    if(empty($_COOKIE['vote_cookie'])==false){          //通过 Cookie 限制重复投票
        echo "<script>alert('您不可以重复投票，请在 1 小时后重新登录！');
            window.location.href='vote.php?vote_subject=".$_GET['more_subject']."';</script>";
    }else{
        setcookie("vote_cookie",'value',time()+3600); //如果不存在，则创建 Cookie
        $k=0;
        while (list($name,$value)=each($_POST)){
            $k+=$name;
        if (is_numeric($name)==true){
    mysql_query("update tb_vote_more set tb_vote_counts=tb_vote_counts+1 where tb_vote_id='".$name."'");
            }
        }
        if ($k==0){
            echo "<script>alert('您不能投票');history.back();</script>";
            exit;
        }
        mysql_query("insert into tb_vote_more_ip(tb_vote_subject,tb_vote_ip)
            values('".$_GET['more_subject']."','$ip')");
        echo "<script>alert('投票提交成功!');
            window.location.href='vote.php?vote_subject=".$_GET['more_subject']."';</script>";
    }
}
?>
```



Note

在执行多选投票内容提交的过程中，使用的是批量添加技术，主要通过 while 循环语句和 list()函数、each()函数来完成。

☑ each()函数：返回数组中当前指针位置的键名和对应的值，并向前移动数组指针。键值对被返回为 4 个单元的数组，键名为 0、1、key 和 value。单元 0 和 key 包含数组单元的键名，1 和 value 包含数据，如果内部指针越过了数组的末端，则函数返回 False。其语法格式如下：

```
array each ( array array)
```

其中，参数 array 为输入的数组。

☑ list()函数：把数组中的值赋给一些变量。与 array()函数类似，不是真正的函数，而是语言结构。list()函数仅能用于数字索引的数组，且数字索引从 0 开始。其语法格式如下：

```
void list ( mixed ...)
```

其中，参数 mixed 为被赋值的变量名称。

说明：

至此，投票内容提交模块介绍完毕。至于投票结果分析的实现方法，参见 8.5.1 与 8.5.2 节，这里不再赘述。

8.4 投 票 管 理

8.4.1 投票管理概述

投票管理是后台主要的模块，控制着前台投票主题和选项以及投票结果。投票管理的功能包括：投票主题、投票内容的添加、浏览和删除，以及投票结果的刷新。投票管理首页的运行效果如图 8.14 所示。



图 8.14 投票管理首页运行效果



8.4.2　动态生成投票主题及选项

投票系统的开发，第一步要做的就是添加投票主题和选项。因为只有具备了投票主题和选项才能够进行投票，否则一切都没有意义。投票主题和选项除了可以从数据库中直接添加以外，还可以创建一个动态添加投票主题和选项的程序，通过动态页实现主题和选项的添加。动态添加投票主题和选项页面的运行效果如图 8.15 所示。



图 8.15　动态生成投票主题及选项

在 admin\vote_content_manage.php 文件中，通过 form 表单将投票的主题和选项添加到对应的数据表中，完成投票主题和选项的动态创建。

（1）创建 form 表单，实现单选或者多选投票选项的动态添加，并且支持图片上传。关键代码如下：

```
<form action="vote_content_manage_ok.php" method="post" enctype="multipart/form-data" name="form2">
<tr bgcolor="#167BE3">
        <td height="25" valign="middle" class="style1"> 投票主题----
<!--从数据表中选择投票的主题-->
        <select name="tb_vote_subject" size="1" id="tb_vote_subject">
<?php
$query="select * from tb_vote_subject";
$result=mysql_query($query);
while($myrow=mysql_fetch_array($result)){
?>
<option value="<?php echo $myrow['tb_vote_subject_name'];?>">
<?php echo $myrow['tb_vote_subject_name'];?></option>
        <?php }?>
      </select></td>
  </tr>
<tr bgcolor="#ECF6FF">
      <td height="30">    
      <input type="submit" name="Submit" value="单选">  
      <input type="submit" name="Submit2" value="多选"></td>
</tr>
```



```
</form>
```

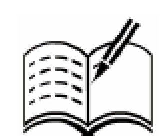
（2）将表单中的数据提交到 vote_content_manage_ok.php 页，根据“提交”按钮的值进行判断，执行将数据添加到单选或者多选的数据表中；并且判断是否上传图片，如果有图片上传则将图片存储到服务器指定的文件夹下。vote_content_manage_ok.php 的关键代码如下：



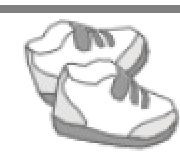
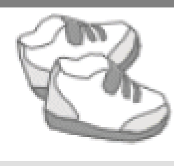
Note

```
<?php
session_start();
include("../conn/conn.php");
if(isset($_SESSION['admin_user']) and isset($_SESSION['admin_pass'])){
    if (isset($_POST['Submit']) && $_POST['Submit']=="单选" and $_FILES['tb_vote_photo']['name']==""){
    //判断是单选，并且没有图片上传
    //执行将数据添加到单选数据表中
$query="insert into tb_vote_odd(tb_vote_subject,tb_vote_content,tb_vote_counts, tb_vote_type,tb_vote_explain)
values('".$_POST['tb_vote_subject']."','".$_POST['tb_vote_content']."','0','".$_POST['Submit']."',
'".$_POST['tb_vote_explain']."')";
    $result=mysql_query($query);
    if ($result){
echo "<script>alert('投票内容提交成功!');window.location.href='index.php?admin_title=管理投票内容';</script>";
    }else{echo "失败!";}
}
if (isset($_POST['Submit']) && $_POST['Submit']=="单选"
and $_FILES['tb_vote_photo']['name']!=""){          //判断是单选，并且有图片上传
    $tb_vote_photo=$_FILES['tb_vote_photo']['size'];  //获取图片大小
    $tb_vote_photo_name=$_FILES['tb_vote_photo']['name']; //获取客户端机器原文件的名称
$tb_vote_photo_type=strtolower(strstr($tb_vote_photo_name,".")); //获取从"."到最后的字符，并将字符转换成小写
    if($tb_vote_photo>2000000){          //判断图片是否超过指定大小
        echo "<script>alert('对不起,您上传的图片超过 2M!');history.back();</script>";
    }else{
        //判断上传图片的后缀名是否符合
        if($tb_vote_photo_type!=".jpg" & $tb_vote_photo_type!=".gif" &
$tb_vote_photo_type!=".bmp" & $tb_vote_photo_type!=".jpeg"){
            echo "<script>alert('对不起,您上传的图片的格式不正确!');history.back();</script>";
        }else{
            //定义图片在服务器中的存储路径
        $tb_vote_photo_path='../images/photo/'.date("YmdHis").mt_rand(1000000,9999999).$_FILES['tb_vote_photo']['name'];
            //应用 move_uploaded_file()函数，将图片存储到指定的文件夹下
            if(move_uploaded_file($_FILES['tb_vote_photo']['tmp_name'],$tb_vote_photo_path)){
            //执行添加语句，将数据添加到单选表中
$query="insert into tb_vote_odd(tb_vote_subject,tb_vote_content,tb_vote_counts,tb_vote_type,tb_
```






```
vote_explain,tb_vote_photo)
values('".$_POST['tb_vote_subject']."','".$_POST['tb_vote_content']."','0','".$_POST['Submit']."',
'".$_POST['tb_vote_explain']."','$tb_vote_photo_path')";
                $result=mysql_query($query);
                if ($result){
echo "<script>alert('投票的内容提交成功!');window.location.href='index.php?admin_title=管理投票内容';</script>";
                }else{
                    echo "失败!";
                }
            }
        }
    }
}
if (isset($_POST['Submit2']) && $_POST['Submit2']=="多选"
and $_FILES['tb_vote_photo']['name']==""){ //判断提交按钮的值是多选，并且无图片上传
$query="insert into tb_vote_more(tb_vote_subject,tb_vote_content,tb_vote_counts,tb_vote_type,tb_vote_explain)
values('".$_POST['tb_vote_subject']."','".$_POST['tb_vote_content']."','0',
'".$_POST['Submit2']."','".$_POST['tb_vote_explain']."')";
    $result=mysql_query($query);
    if ($result){
echo "<script>alert('投票内容提交成功!');window.location.href='index.php?admin_title=管理投票内容';</script>";
    }else{echo "失败!";}
}
if (isset($_POST['Submit2']) && $_POST['Submit2']=="多选" and $_FILES['tb_vote_photo']['name']!=""){
    $tb_vote_photo=$_FILES['tb_vote_photo']['size'];
    $tb_vote_photo_name=$_FILES['tb_vote_photo']['name']; //获取客户端机器原文件的名称
$tb_vote_photo_type=strtolower(strstr($tb_vote_photo_name,".")); //获取从"."到最后的字符，并将字符转换成小写
    if($tb_vote_photo>2000000){
        echo "<script>alert('对不起,您上传的图片超过 2M!');history.back();</script>";
    }else{
        if($tb_vote_photo_type!=".jpg" & $tb_vote_photo_type!=".gif" & $tb_vote_photo_type!=".bmp"){
            echo "<script>alert('对不起,您上传的图片的格式不正确!');history.back();</script>";
        }else{
    $tb_vote_photo_path='../images/photo/'.date("YmdHis").mt_rand(1000000,9999999).$_FILES['tb_vote_photo']['name'];
            if(move_uploaded_file($_FILES['tb_vote_photo']['tmp_name'],$tb_vote_photo_path)){
$query="insert into tb_vote_more(tb_vote_subject,tb_vote_content,tb_vote_counts,tb_vote_type,tb_vote_explain,tb_vote_photo)
values('".$_POST['tb_vote_subject']."','".$_POST['tb_vote_content']."','0','".$_POST
```

```
['Submit2']."',
    '".$_POST['tb_vote_explain']."','$tb_vote_photo_path')";
                    $result=mysql_query($query);
                    if ($result){
echo "<script>alert('投票内容提交成功!');window.location.href='index.php?admin_title=管理投票内容';</script>";
                    }else{
                        echo "失败!";
                    }
                }
            }
        }
    }
    }else{
        echo "<script>alert('您没有正确登录！ ');window.location.href='admin_enter.php'; </script>";
    }
    ?>
```

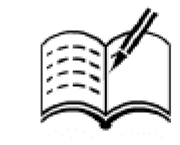
8.4.3 删除投票主题与投票内容

在图 8.15 中，单击“删除”按钮，即可删除当前投票主题和选项。删除操作的执行，是根据指定投票主题的 ID，执行 delete 删除语句。删除投票主题的同时，也将删除此主题下的选项内容。这一操作是在 delete_vote_subject.php 文件中执行的，其关键代码如下：

```
<?php
session_start();
include("../conn/conn.php");
if(isset($_SESSION['admin_user']) && isset($_GET['vote_subject'])){
    $query1=mysql_query("delete from tb_vote_subject where tb_vote_subject_name='$_GET[vote_subject]'");
    $query2=mysql_query("delete from tb_vote_odd where tb_vote_subject='$_GET[vote_subject]'");
    $query3=mysql_query("delete from tb_vote_odd_ip where tb_vote_subject='$_GET[vote_subject]'");
    $query4=mysql_query("delete from tb_vote_more where tb_vote_subject='$_GET[vote_subject]'");
    $query5=mysql_query("delete from tb_vote_more_ip where tb_vote_subject='$_GET[vote_subject]'");
    echo "<script>alert('删除成功!'); window.location.href='index.php?admin_title=管理投票主题';</script>";
}else{
    echo "<script>alert('请您正确登录!'); window.location.href='admin_enter.php'; </script>";
}
?>
```

如果用户只是想删除投票的某一项内容，可以在图 8.15 中单击“管理投票内容”超链接，在显示的投票选项中找到需要删除的选项内容，单击“删除”超链接，即可将其删除。





这一操作是在 delete_vote_content.php 文件中执行的，其关键代码如下。

```
<?php
session_start();
include("../conn/conn.php");
if(isset($_SESSION['admin_user'])){
	if(isset($_GET['vote_content'])){
		$query2=mysql_query("select * from tb_vote_odd where tb_vote_content='".$_GET['vote_content']."'");
		$myrow=mysql_fetch_array($query2);
		if($myrow['tb_vote_photo']==true){
			unlink($myrow['tb_vote_photo']);
		}
		$query="delete from tb_vote_odd where tb_vote_content='".$_GET['vote_content']."'";
		$result=mysql_query($query);
	echo "<script>alert('删除成功!');window.location.href='index.php?admin_title=管理投票内容';</script>";
	}
	if(isset($_GET['vote_contents'])){
	$query4=mysql_query("select * from tb_vote_more where tb_vote_content='".$_GET['vote_contents']."'");
		$myrow4=mysql_fetch_array($query4);
		if($myrow4['tb_vote_photo']==true){
			unlink($myrow4['tb_vote_photo']);
		}
		$querys="delete from tb_vote_more where tb_vote_content='".$_GET['vote_contents']."'";
		$results=mysql_query($querys);
	echo "<script>alert('删除成功!');window.location.href='index.php?admin_title=管理投票内容';</script>";
	}
}else{
	echo "<script>alert('请您正确登录!'); window.location.href='admin_enter.php';</script>";
}
?>
```

8.4.4　刷新投票结果

刷新投票就是将当前的投票结果清零，重新开始投票。在图 8.16 所示的页面中，单击“重新投票”超链接，将跳转到 update_vote.php 文件中，根据超链接传递的参数值，完成对指定投票选项的更新操作。

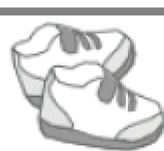
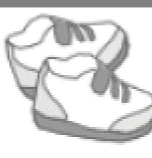






图 8.16　投票系统管理

下面讲解在投票管理模块中刷新投票结果的方法。其关键是在 update_vote.php 文件中，根据超链接中传递的 ID 值，将指定数据表中的 tb_vote_counts 字段的值更新为 0，并删除指定数据表中对应的 IP 地址的记录。关键代码如下：

```
<?php
session_start();
include("../conn/conn.php");
if(isset($_SESSION['admin_user'])){
        if(isset($_GET['type_odd']) && $_GET['type_odd']=="单选"){
$query=mysql_query("update tb_vote_odd set tb_vote_counts='0' where tb_vote_subject='".$_GET['update_id']."'");
                $querys=mysql_query("delete from tb_vote_odd_ip where tb_vote_subject='".$_GET['update_id']."'");
                echo "<script>alert('刷新成功！');
window.location.href='index.php?admin_title=管理投票&&vote_subjectes=".$_GET['update_id']."';</script>";
        }
        if(isset($_GET['type_more']) && $_GET['type_more']=="多选" ){
$query=mysql_query("update tb_vote_more set tb_vote_counts='0' where tb_vote_subject='".$_GET['update_id']."'");
                $querys=mysql_query("delete from tb_vote_more_ip where tb_vote_subject='".$_GET['update_id']."'");
                echo "<script>alert('刷新成功！');
window.location.href='index.php?admin_title=管理投票&&vote_subjectes=".$_GET['update_id']."';</script>";
        }
}else{
        echo "<script>alert('请您正确登录!'); window.location.href='admin_enter.php';</script>";
}
?>
```





8.5　技　术　提　炼

8.5.1　通过 3D 饼图分析投票结果

为了更好地展示在线投票的结果，使其更加直观，下面介绍一种通过 3D 饼图来分析投票结果的方法。其运行效果如图 8.17 所示。

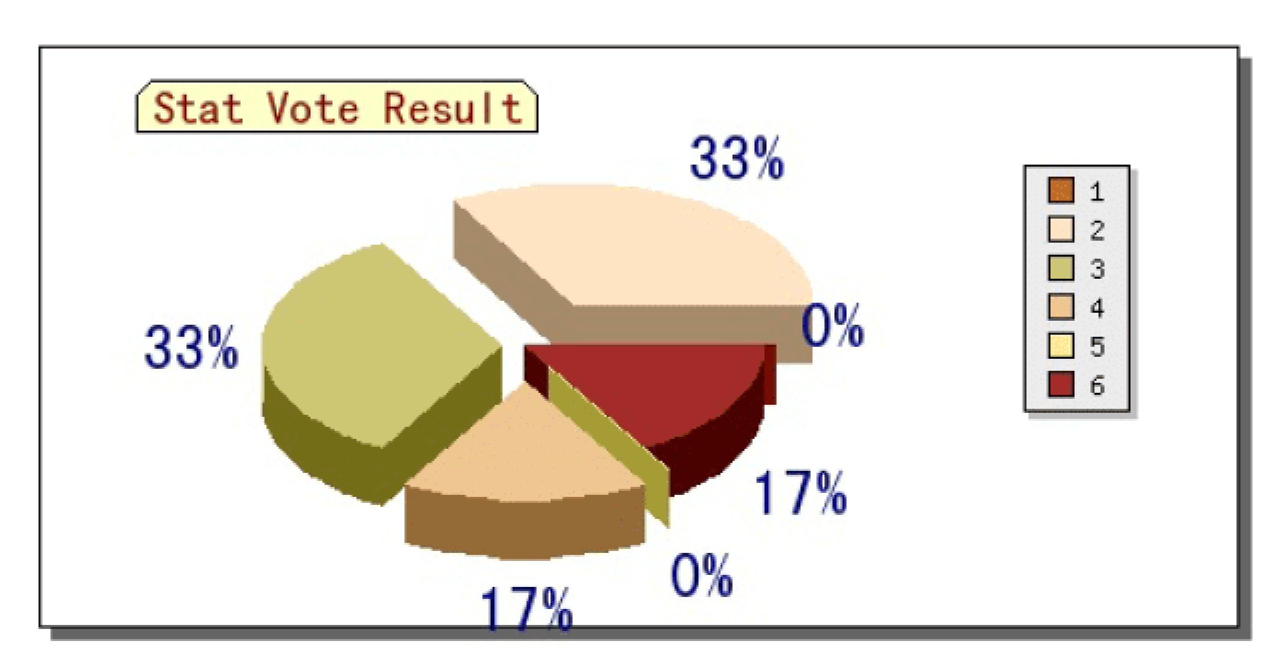


图 8.17　通过 3D 饼图分析投票结果

通过 3D 饼图分析投票结果，其关键就是 3D 饼图的创建和如何将投票结果的数据传递到 3D 饼图中。3D 饼图的创建使用的是 jpgraph 类库。在讲解饼图的创建之前，先来了解一下 jpgraph 类库。

jpgraph 是一个强大的绘图组件，能根据用户的需要绘制任意图形。只要提供数据，它就能自动调用绘图函数把处理的数据填进去自动绘制。jpgraph 类库提供了多种方法类创建各种统计图，包括折线图、柱形图和饼图等。jpgraph 是一个完全使用 PHP 语言编写的类库，可以应用在任何 PHP 环境中。

说明：

jpgraph 需要 GD 库的支持。如果用户希望 jpgraph 类库仅对当前站点有效，可将 jpgraph 压缩包下的 src 文件夹中的全部文件复制到网站所在目录的文件夹中即可，在使用时调用 src 文件夹下的指定文件即可。

下面介绍在 see_vote_results.php 文件中，如何应用 Jpgraph 创建 3D 饼图，对投票系统多选投票的结果进行分析。其操作步骤如下：

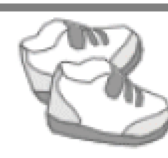
（1）连接数据库。

（2）应用 include_once 语句引用指定的文件，包括 jpgraph.php、jpgraph_pie.php 和 jpgraph_ pie3d.php 文件。其中 jpgraph_pie.php 文件是饼图对象所在的文件，jpgraph_pie3d.php 是 3D 饼图 PiePlot3D 对象所在的类文件。

（3）编写 PHP 语句，从数据库中统计出投票数量。

（4）将获取的投票数据写入一个数组中。

（5）创建 Graph 对象，生成一个 540 像素×260 像素大小的画布。





Note

（6）设置标题的字体以及图例的字体。

（7）定义生成图像在画布中的位置。

（8）将绘制的 3D 饼图添加到图像中。

（9）最后输出图像。

关键代码如下：

```
<?php
include("conn/conn.php");
include ("src/jpgraph.php");
include ("src/jpgraph_pie.php");
include ("src/jpgraph_pie3d.php");              //引用 3D 饼图 PiePlot3D 对象所在的类文件
//统计投票的总量
$query=mysql_query("select sum(tb_vote_counts) as vote_gross from tb_vote_more
where tb_vote_subject='".$_GET['vote_subject']."'");
$vote_counts=mysql_result($query,0,"vote_gross");       //获取总的投票数量
if($vote_counts>0){
    $querys=mysql_query("select * from tb_vote_more where tb_vote_subject='".$_GET
['vote_subject']."' ");
    $vote_content=array();
    $resultes=array();
    while($myrow=mysql_fetch_array($querys)){
        $vote_content[]=$myrow['tb_vote_id'];
        $resultes[]=$myrow['tb_vote_counts'];           //将获取的值写入到数组中
    }
$graph = new PieGraph(500,245,"auto");                  //创建图像
$graph->SetShadow();                                    //创建图像阴影
$graph->tabtitle->Set('Stat Vote Result');              //输出标题
$graph->tabtitle->SetFont(FF_SIMSUN, FS_BOLD,14);       //设置标题字体
$graph->title->SetColor("darkblue");                    //定义标题颜色
$graph->legend->Pos(0.1,0.2);                           //控制注释文字的位置
$p1 = new PiePlot3d($resultes);                         //创建图像
$p1->SetTheme("sand");                                  //控制图像颜色
$p1->SetCenter(0.4);                                    //设置图像位置
$p1->SetSize(0.4);                                      //设置图像大小
$p1->SetHeight(20);                                     //设置饼图高度
$p1->SetAngle(45);                                      //设置图像倾斜角度
$p1->Explode(array(5,40,10,30,20));                     //控制饼图的分割
$p1->value->SetFont(FF_SIMSUN, FS_BOLD,20);             //设置字体
$p1->SetLegends($vote_content);
$graph->Add($p1);                                       //添加数据
$graph->Stroke();                                       //生成图像
}
?>
```

8.5.2 通过柱形图分析投票结果

对投票结果的分析，不但可以通过 3D 饼图来完成，也可以使用柱形图来分析。下面介绍如何通过柱形图对投票系统单选投票的结果进行分析，其运行效果如图 8.18 所示。



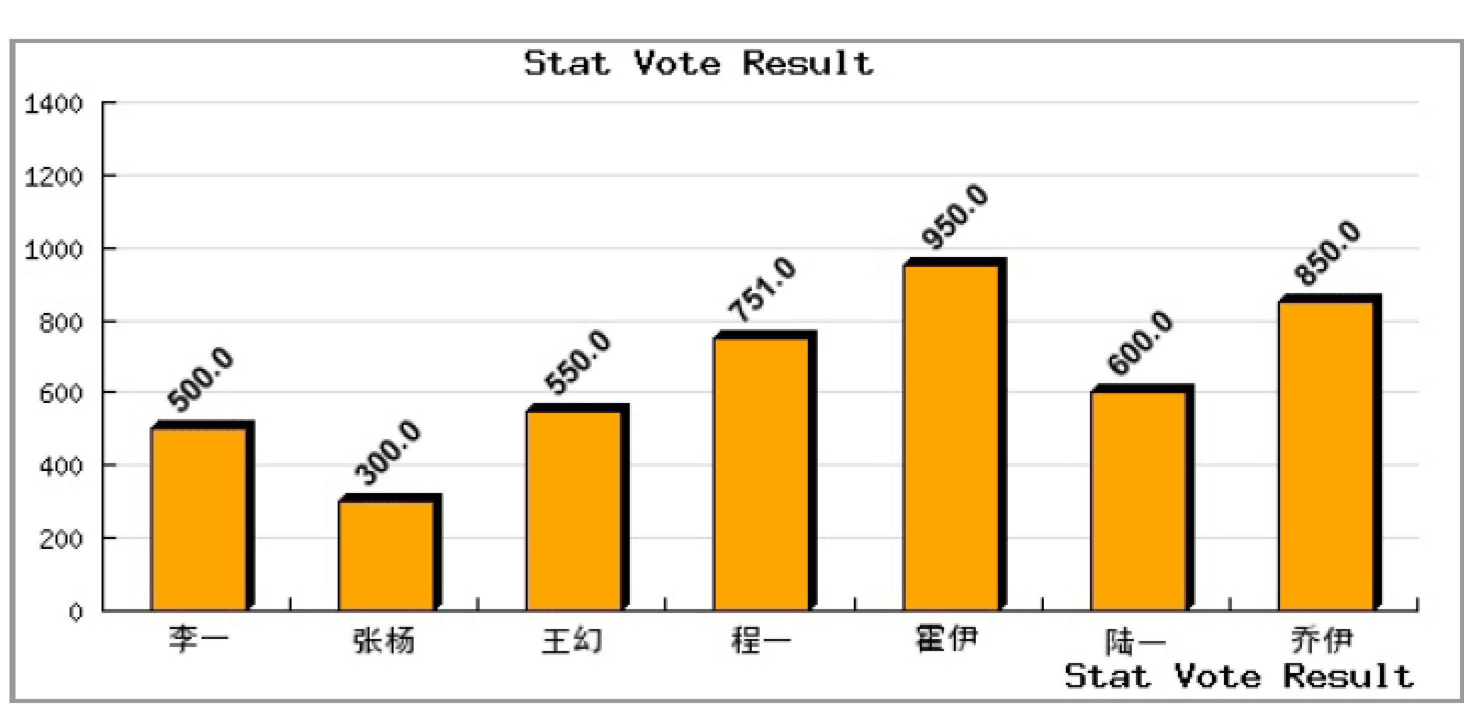
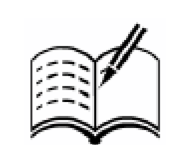


图 8.18　通过柱形图分析投票结果



创建 see_vote_result.php 文件，载入数据库连接文件，载入 jpgraph 类库，完成柱形图的创建。其步骤如下：

（1）通过 include 语句引用 conn.php 和 jpgraph.php 文件。

（2）应用 inlcude 语句调用 jpgraph_bar.php 文件，用于创建 BarPlot 对象。

（3）从数据库中读取出投票总量，计算每项投票数量在总的投票数量中所占的比例，并将计算结果写入一个数组中。

（4）创建 Graph 对象，生成一个 600 像素×300 像素大小的画布。定义生成图像在画布中的位置，以及画布的阴影、蓝色背景等。

（5）创建一个矩形对象 BarPlot，设置其柱形图的颜色，在柱形图上方显示投票数据，并格式化数据为整型。

（6）将绘制的柱形图添加到画布中。

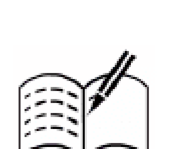
（7）添加标题名称和 X 轴坐标，并分别设置其字体。

（8）最后输出图像。

关键代码如下：

```
<?php
include("conn/conn.php");
include ("src/jpgraph.php");
include ("src/jpgraph_bar.php");
//统计投票的总量
$query=mysql_query("select sum(tb_vote_counts) as vote_gross from tb_vote_odd
where tb_vote_subject='".$_GET['vote_subject']."'");
$vote_counts=mysql_result($query,0,"vote_gross");        //获取总的投票数量
if($vote_counts>0){
    $querys=mysql_query("select * from tb_vote_odd where tb_vote_subject='".$_GET['vote_subject']."' ");
    $vote_content=array();
    $resultes=array();
    while($myrow=mysql_fetch_array($querys)){
        $vote_content[]=$myrow['tb_vote_content'];
        $results=number_format($myrow['tb_vote_counts']/1,2); //计算每项投票数量占总投票量的比例
        $resultes[]=$results;                            //将获取的值写入到数组中
```





Note

```
        }
$datay=$resultes;
$graph = new Graph(650,280,'auto');                    //创建图像
$graph->img->SetMargin(80,30,30,40);                    //定义图像边框间距
$graph->SetScale("textint");                            //定义刻度值的类型
$graph->SetShadow();                                    //设置图像阴影
$graph->SetFrame(false);                                //不设置图像的背景
$graph->yaxis->scale->SetGrace(35);                     //设置 Y 轴距离顶部的距离
$graph->xaxis->SetTickLabels($vote_content);            //添加数据
$graph->xaxis->SetFont(FF_SIMSUN, FS_BOLD);             //设置字体
$graph->title->Set('Stat Vote Result');                 //定义标题
$graph->title->SetFont(FF_FONT2,FS_BOLD);               //设置标题字体
$graph->xaxis->title->Set("Stat Vote Result");          //设置 X 轴的坐标
$graph->xaxis->title->SetFont(FF_FONT2,FS_BOLD);        //定义字体
$bplot = new BarPlot($datay);                           //创建图像
$bplot->SetFillColor("orange");                         //定义图像的填充颜色
$bplot->SetWidth(0.5);                                  //设置图像的大小
$bplot->SetShadow();                                    //设置图像的阴影
$bplot->value->Show();                                  //输出图像对应的数据值
$bplot->value->SetFont(FF_ARIAL,FS_BOLD);               //定义图像值的字体
$bplot->value->SetAngle(45);                            //定义图像值的输出角度
//$bplot->value->SetFormat('$ %0.0f');                  //对输出值进行格式化
$bplot->value->SetColor("black","darkred");             //定义图像值的颜色
$graph->Add($bplot);                                    //添加数据
$graph->Stroke();                                       //生成图像
}
?>
```

8.5.3 通过 Cookie 控制重复投票

在投票系统中有一个很关键的问题，就是重复投票。应该通过什么方法来处理这个问题？投票系统是否允许重复投票？……对此解决方案很多，关键是根据系统的实际使用情况而定，是想要控制一个 IP 地址只进行一次投票，还是每间隔一段时间进行一次投票，或者是在重新登录时进行一次投票。

下面介绍一种通过 Cookie 来控制重复投票的方法。该方案实现了每间隔一段时间进行一次投票，即同一个 IP 地址可以进行多次投票，只是在多次投票之间有一个间隔的时间。其运行效果如图 8.19 所示。

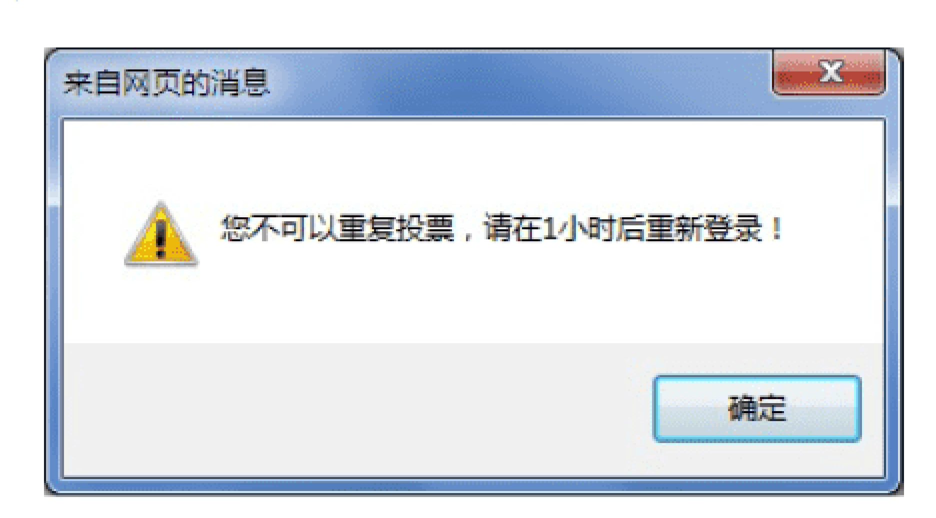
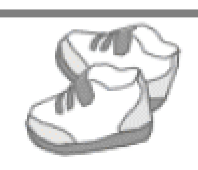


图 8.19　通过 Cookie 控制重复投票



Note

通过 Cookie 控制重复投票的原理是：当用户登录到投票系统时，系统将判断当前客户端的 Cookie 变量的值是否为空，如果 Cookie 变量不为空则不能进行投票；如果为空则可以进行投票，并且通过 setcookie()函数创建一个 Cookie 变量，指定 Cookie 的过期时间为 1 小时，以此来限制当前客户端在 1 小时之内不能重复投票。在投票系统的 tb_vote_ok.php 文件中，通过 Cookie 对多选投票进行限制，限制其在 1 小时之内不能进行重复投票。其关键代码如下：

```
if (isset($_POST['Submit2'])){
    if(empty($_COOKIE['vote_cookie'])==false){          //通过 Cookie 控制重复投票
        echo "<script>alert('您不可以重复投票，请在 1 小时后重新登录！');
window.location.href='vote.php?vote_subject=".$_GET['more_subject']."';</script>";
    }else{
        setcookie("vote_cookie",'value',time()+3600); //如果不存在，则创建 Cookie
        $k=0;
    while (list($name,$value)=each($_POST)){
            $k+=$name;
        if (is_numeric($name)==true){
 mysql_query("update tb_vote_more set tb_vote_counts=tb_vote_counts+1 where tb_vote_id='".$name."'");
            }
        }
        if ($k==0){
            echo "<script>alert('您不能投票');history.back();</script>";
            exit;
        }
mysql_query("insert into tb_vote_more_ip(tb_vote_subject,tb_vote_ip)values('".$_GET['more_subject']."','$ip')");
echo "<script>alert('投票提交成功!');
window.location.href='vote.php?vote_subject=".$_GET['more_subject']."';</script>";
    }
}
```

8.5.4　通过 IP 限制重复投票

Cookie 和 Session 都是对投票的时间间隔进行控制，确切地说没有完全对重复投票进行限制。这里介绍一种完全限制重复投票的方法，即通过 IP 限制重复投票。其运行效果如图 8.20 所示。

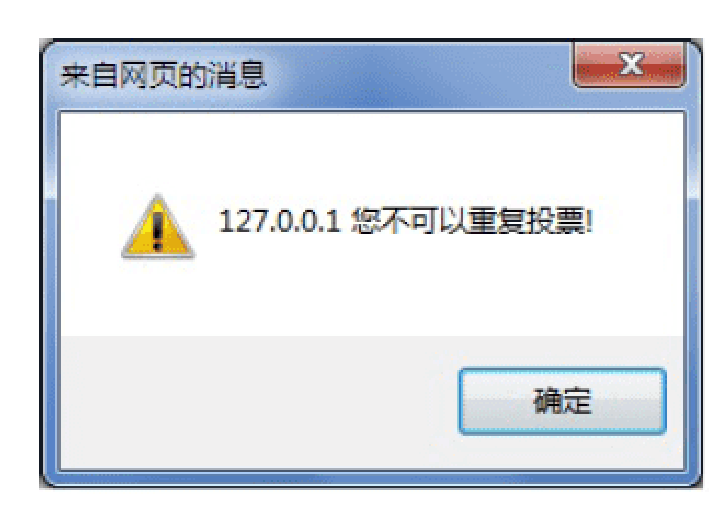
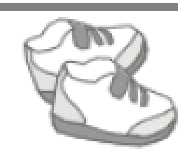
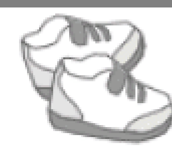


图 8.20　通过 IP 限制重复投票



Note

通过 IP 地址限制重复投票的原理是：当用户提交投票时，首先获取到客户端计算机的 IP 地址，然后判断在指定的数据表中是否存在该 IP 地址，如果存在该 IP 地址，则说明该 IP 已经进行过投票，提示“您不可以重复投票!”；否则，可以进行投票，并且将 IP 地址记录到指定的数据表中。在投票系统的 tb_vote_ok.php 文件中，应用 IP 地址对单选投票进行限制。关键代码如下：

```
<?php
session_start();
include("conn/conn.php");
$data=date("Y-m-d h:i:s");
$ip=getenv("REMOTE_ADDR");                    //获取 IP
if (isset($_POST['Submit'])){
    $tb_vote_id=$_POST['tb_vote_id'];
    if ($_POST['tb_vote_id']==""){
        echo "<script>alert('您没有选择投票的内容!');
window.location.href='vote.php?vote_subject=".$_POST['tb_vote_subject']."';</script>";
    }else{
$query="select * from tb_vote_odd_ip where tb_vote_subject='".$_POST['tb_vote_subject']."' and tb_vote_ip='$ip'";
        $result=mysql_query($query);
        $row=mysql_num_rows($result);
        if ($row>0){           //判断此 IP 是否重复
            echo "<script>alert('".$ip." 您不可以重复投票!');
window.location.href='vote.php?vote_subject=".$_POST['tb_vote_subject']."';</script>";
        }else{
    $querys="update tb_vote_odd set tb_vote_counts=tb_vote_counts+1 where tb_vote_id='$tb_vote_id'";
            $result=mysql_query($querys);
            $query=mysql_query("select * from tb_vote_odd where tb_vote_id='$tb_vote_id'");
            $myrow=mysql_fetch_array($query);
            $tb_vote_subject=$myrow['tb_vote_subject'];
            $tb_vote_content=$myrow['tb_vote_content'];
$querys="insert into tb_vote_odd_ip(tb_vote_subject,tb_vote_content,tb_vote_ip,tb_vote_date,tb_vote_id )
values('$tb_vote_subject','$tb_vote_content','$ip','$data','$tb_vote_id')";
            $result=mysql_query($querys);
            echo "<script>alert('投票提交成功!');
window.location.href='vote.php?vote_subject=".$_POST['tb_vote_subject']."';</script>";
        }
    }
}
```

技巧：

在通过 IP 限制重复投票的过程中，客户端 IP 地址的获取使用的是 getenv()函数。getenv()函数主要是用于获取环境变量的值，即客户端的 IP 地址。此外，也可以使用$_SERVER['REMOTE_ADDR']来获取客户端的 IP 地址。

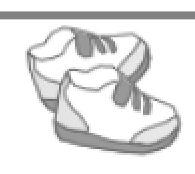
第9章

留言本模块

（自学视频、源程序：配套资源\mr\9\）

本章以构建一个适用于中小型企业的留言本为目的，全面介绍留言本的设计思路及流程。在介绍留言本开发的同时，还会对PHP中一些常用函数及编程技巧进行详细介绍。这样，不仅可以使读者对留言本的开发过程及思路做到全面了解并掌握，而且还可以学习到许多PHP的编程技巧，从而全面提高个人编程能力。留言本的开发过程虽然较为简单，却涵盖了动态网站开发的大部分经典功能模块，所以只要熟练掌握本章的内容，并且能够灵活应用，就可以开发出大型的网上论坛、博客以及其他一些网络中较流行的网站。通过本章的学习，读者可以学到以下内容。

- 掌握留言本的设计思路及流程
- 掌握MySQL数据库的设计方式和方法
- 掌握PHP连接MySQL数据库的方法
- 能够灵活运用常用的SQL查询语句
- 掌握PHP实现数据分页显示的方法





Note

9.1 留言本模块概述

无论是应用程序还是 Web 项目，开发人员只有提前作好规划并清楚地了解其设计思路及流程，才能高效、高质量地完成开发。本节将从留言本的功能概述、功能结构等几方面介绍留言本的设计思路及实现过程。

9.1.1 模块概述

开发一个适合个人或小型企业使用的留言本，只需提供添加留言和删除留言功能即可。由于本章介绍的留言本是以适用于中小型企业为出发点，目的是使读者全面掌握留言本的设计流程及提高个人的编程能力，所以除了具有添加留言和删除留言的功能外，还提供了用户注册、用户登录、编辑留言等功能。本章所介绍的留言本运行效果如图 9.1 所示。



图 9.1 留言本运行效果

9.1.2 功能结构

在开发程序前，先规划一下留言本模块的功能结构，以便在开发程序时保证思路清晰。留言本模块的功能结构如图 9.2 所示。



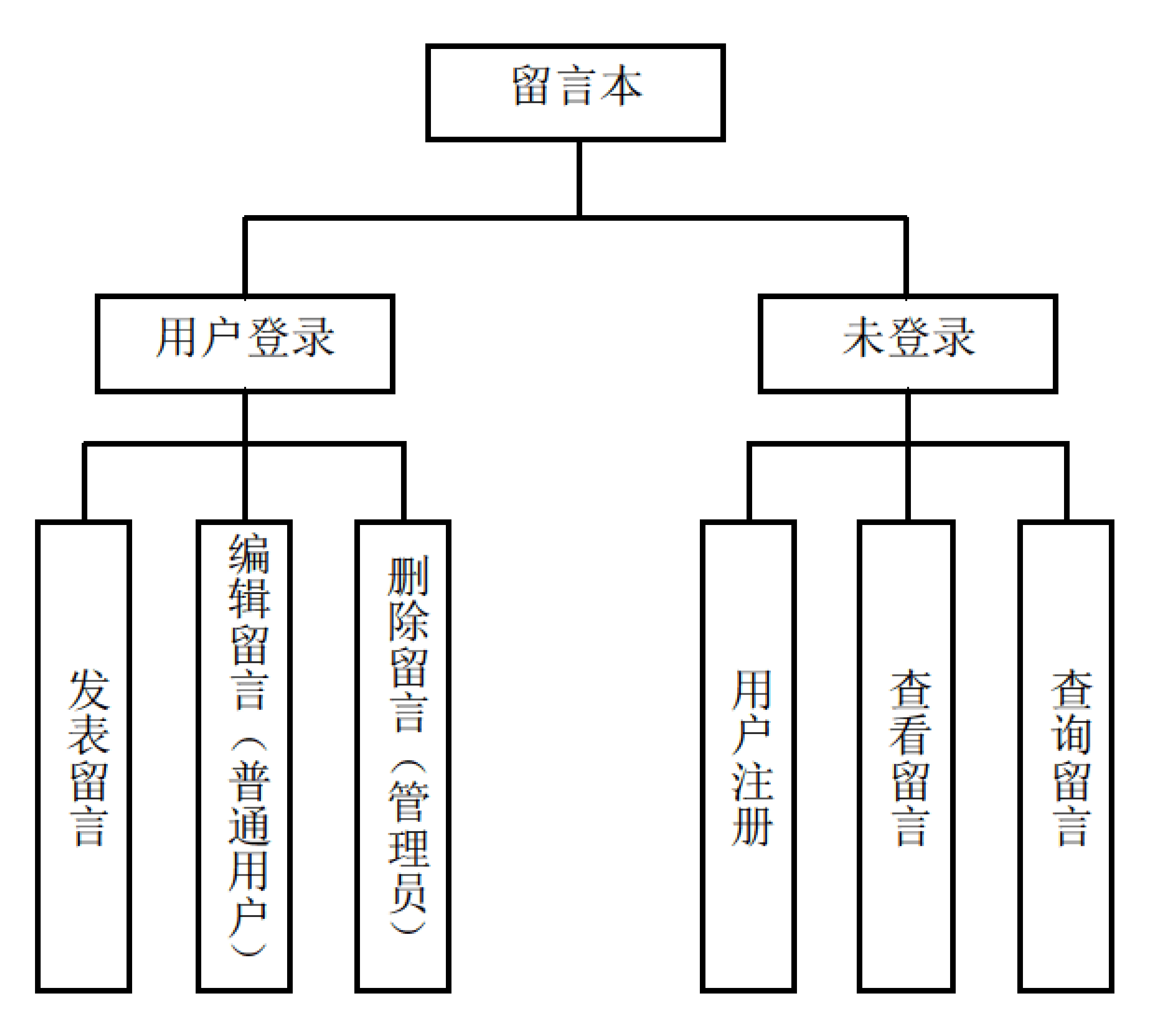



图 9.2　留言本模块的功能结构图

9.1.3　程序预览

留言本是由多个功能模块组成的，为了让读者对本系统有个初步的了解和认识，下面列出几个典型功能模块的运行效果图，其他参见配套资源中的源程序。

首页运行效果如图 9.3 所示。



图 9.3　留言本首页

Note




Note

用户注册页面运行效果如图 9.4 所示。用户只有注册成功后才可以获取登录用户名和密码；进行登录后，才能执行发表留言等操作。



图 9.4　用户注册页面

发表留言页面运行效果如图 9.5 所示；查找留言页面运行效果如图 9.6 所示。

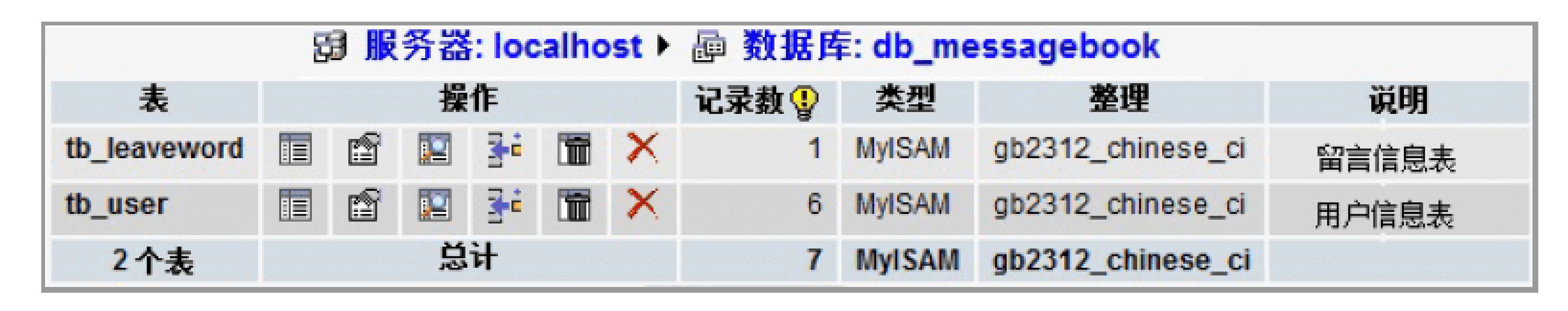


图 9.5　发表留言页面　　　　图 9.6　查找留言页面

9.2　数据库设计

9.2.1　数据库设计

留言本模块采用的是 MySQL 数据库，主要用于存储用户信息和留言信息。这里将数据库命名为 db_ messagebook，其中包含的数据表如图 9.7 所示。



服务器: localhost ▸ 数据库: db_messagebook

表	操作	记录数	类型	整理	说明
tb_leaveword		1	MyISAM	gb2312_chinese_ci	留言信息表
tb_user		6	MyISAM	gb2312_chinese_ci	用户信息表
2 个表	总计	7	MyISAM	gb2312_chinese_ci	

图 9.7　数据库结构



9.2.2　数据表设计

数据库设计完成后，下面来看各个数据表的结构和字段说明。

tb_user（用户信息表）主要用于存储用户信息，其结构如图 9.8 所示。

服务器: localhost ▸ 数据库: db_messagebook ▸ 表 : tb_user

字段	类型	整理	属性	Null	默认	额外	说明
id	int(8)			否		auto_increment	自动编号
usernc	varchar(50)	gb2312_chinese_ci		否			用户名称
truename	varchar(50)	gb2312_chinese_ci		否			真实名称
email	varchar(50)	gb2312_chinese_ci		否			邮箱
qq	varchar(20)	gb2312_chinese_ci		否			QQ
tel	varchar(20)	gb2312_chinese_ci		否			电话
ip	varchar(10)	gb2312_chinese_ci		否			IP
address	varchar(250)	gb2312_chinese_ci		否			地址
face	varchar(50)	gb2312_chinese_ci		否			头像
regtime	datetime			否			发布时间
sex	varchar(2)	gb2312_chinese_ci		否			性别
usertype	int(2)			否			权限
userpwd	varchar(50)	gb2312_chinese_ci		否			登录密码

图 9.8　用户信息表

tb_leaveword（留言信息表）主要用于存储用户留言信息，其结构如图 9.9 所示。

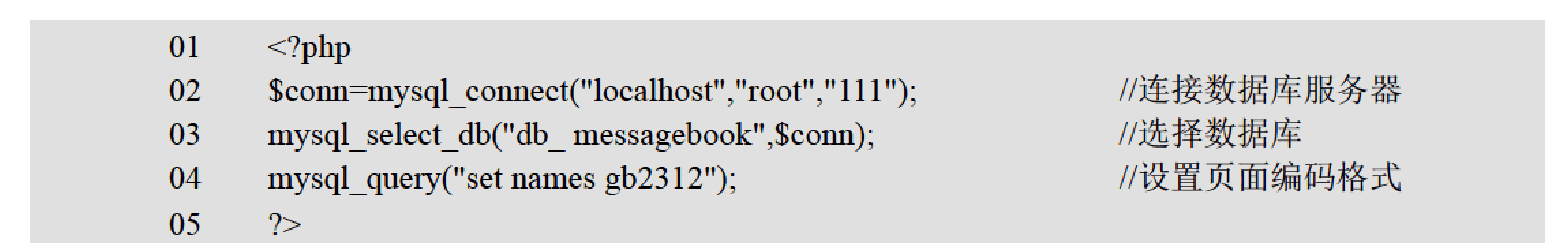
服务器: localhost ▸ 数据库: db_messagebook ▸ 表 : tb_leaveword

字段	类型	整理	属性	Null	默认	额外	说明
id	int(8)			否		auto_increment	留言id
userid	int(8)			否			留言者id
createtime	datetime			否			留言时间
title	varchar(250)	gb2312_chinese_ci		否			留言主题
content	text	gb2312_chinese_ci		否			留言内容

图 9.9　留言信息表

9.2.3　连接数据库

由于大部分页面都需要使用数据库，如果每页都编写相同的数据库连接代码，会显得十分繁琐，所以本模块将数据库连接代码单独存入一个 PHP 文件 conn.php 中，在需要与数据库连接的页面中，使用包含语句包含 conn.php 文件即可。该模块实现与数据库连接的代码如下：



```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_ messagebook",$conn);               //选择数据库
mysql_query("set names gb2312");                        //设置页面编码格式
?>
```

9.3　发表留言

9.3.1　发表留言概述

用户登录成功后，单击导航栏中的“发表留言”超链接，即可进入发表留言页面，如

图 9.10 所示。在该页面中填写留言主题及留言内容，然后单击“发表”按钮，即可发表留言。



图 9.10 发表留言页面

9.3.2 发表留言页面设计

一个优秀的 Web 程序，不仅应具有合理的代码编写规则和较高的代码执行效率，合理的页面设计方式和美观的页面效果也是不可缺少的。为了保证整个留言本页面的一致性，在设计页面时，将留言本的头部内容存储在 top.php 文件中，将用于显示版权信息的尾部内容存储在 bottom.php 文件中。这样，在新建留言本的功能页面时，只需在页面最上方加上 include_once("top.php")和在页面最下方加上 include_once("bottom.php")即可。应用这种页面设计方式，还可以提高程序的开发效率和易维护性。

技巧：

为了提高网络的传输速度，应尽量将页面图片存储为 GIF 或 JPG 格式（这两种格式的图片具有占用空间小、画面质量高等特点）。另外，美观得体的页面效果也是开发人员所必须考虑的因素之一。

发表留言页面的设计效果如图 9.11 所示。





图 9.11 发表留言页面的设计效果

发表留言页面的设计流程如下：

（1）应用 include 语句引用顶部 Banner 广告头文件 top.php，代码如下：

```
<?php include("top.php"); ?>
```

（2）应用 include 语句引用左侧功能导航文件 left.php，代码如下：

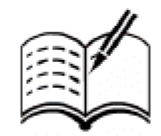
```
<?php include("left.php"); ?>
```

（3）为了使页面效果更丰富多彩、更人性化，分别定义文本框和按钮的 CSS 样式来制作个性化的表单。文本框和按钮的 CSS 样式代码如下：

```
<--***********定义按钮的 CSS 样式*************->
buttoncss {
	font-family: "Tahoma", "宋体";
	font-size: 9pt; color: #FCC42C;
	border: 1px #003399 solid;
	color:006699;
	BORDER-BOTTOM: #FCC42C 1px solid;
	BORDER-LEFT: #FCC42C 1px solid;
	BORDER-RIGHT: #FCC42C 1px solid;
	BORDER-TOP: #FCC42C 1px solid;
	background-color: #ffffff;
	CURSOR: hand;
	font-style: normal ;
}
<--************定义文本框的 CSS 样式**********-->
.inputcss {
	font-size: 9pt;
```

```
        font-family: "宋体";
        font-style: normal;

    }
```


Note

发表留言页面涉及到的 HTML 表单中的重要元素如表 9.1 所示。

表 9.1　发表留言页面涉及到的 HTML 表单中的重要元素

名　称	类　型	含　义	重要属性
form1	form	表单	name="form1"　method="post"　action="saveleaveword.php" onsubmit="return chkinput(this)"
title	text	留言主题	type="text" name="title" size="40" class="inputcss"
content	textarea	留言内容	name="content" rows="15" cols="55" class="inputcss"
submit	submit	“发表”按钮	type="submit" name="submit" value="发表" class="buttoncss"
reset	reset	“重写”按钮	type="reset" name="reset" value="重写" class="buttoncss"

9.3.3　将用户留言内容保存到数据库中

要实现发表留言功能，首先应建立用于填写留言信息的表单，然后通过 PHP 脚本获取表单提交的数据，将留言信息添加到指定的数据表中。在 leaveword.php 中完成填写留言信息表单的创建，在 saveleaveword.php 文件中获取表单提交的留言信息，并且将其添加到指定的数据表中。其关键代码如下：

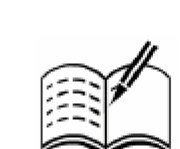
```
<?php
session_start();                              //初始化 SESSION 变量
include_once("conn.php");                     //包含数据库连接文件
$sql=mysql_query("select id from tb_user where usernc='".$_SESSION["unc"]."'",$conn);    //执行查询语句
$info=mysql_fetch_array($sql);                //获取查询结果
$userid=$info['id'];                          //获取用户 ID
$createtime=date("Y-m-d H:i:s");              //获取系统当前时间
//执行添加语句
if(mysql_query("insert into tb_leaveword(userid,createtime,title,content)
values('$userid','$createtime','".$_POST['title']."','".$_POST['content']."')",$conn)){
        echo "<script>alert('留言发表成功！');history.back();</script>";
}else{
        echo "<script>alert('留言发表失败！');history.back();</script>";
}
?>
```

9.4　查 看 留 言

9.4.1　查看留言概述

单击导航栏中的“查看留言”超链接，即可进入查看留言页面，如图 9.12 所示。在该





页面中，可以查看留言主题、内容以及发布者等信息，还可对当前登录用户的权限进行判断，如果当前用户是留言的发布者，那么还可以对留言进行编辑，否则只能查看留言内容。



图 9.12　查看留言页面

9.4.2　查看留言页面设计

为了使留言者和留言内容相互对应，在设计查看留言页面时，将留言者的常规信息动态显示在页面左侧，将留言内容显示在右侧，并通过自定义函数对用户留言内容中的 HTML 标记进行过滤。查看留言页面的设计效果如图 9.13 所示。

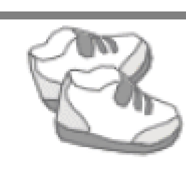


图 9.13　查看留言页面的设计效果

查看留言页面的设计流程如下：

（1）在 HTML 标记中嵌入 PHP 代码，动态地显示用户常规信息。代码如下：

```
<img src="images/email.gif" width="45" height="16" alt="<?php echo $info1[email];?>"/>
<img src="images/ip.gif" width="55" height="16"   alt="<?php echo $info1[ip];?>"/>
<img src="images/qq.gif" width="45" height="16"   alt="<?php echo $info1[qq];?>"/>
```



（2）通过自定义函数 unhtml()实现用户留言中的 HTML 标记原样输出。代码如下：

```
function unhtml($content){
    $content=htmlspecialchars($content);            //将单引号、小于号、大于号等原样输出
    $content=str_replace(chr(13),"<br>",$content);        //原样输出换行标记
    $content=str_replace(chr(32)," ",$content);      //原样输出空格
  return trim($content);
 }
```

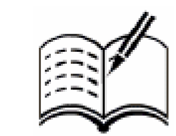
9.4.3 在页面中输出留言信息

在 index.php 文件中，通过 while 语句循环输出留言信息的内容，并且应用分页技术控制每页显示 3 条记录。其关键代码如下：

```
<?php
    $sql=mysql_query("select count(*) as total from tb_leaveword ",$conn);
    $info=mysql_fetch_array($sql);
    $total=$info[total];                          //获取总留言条数
    if($total==0){                                //如果总留言条数为 0，则给出提示
     echo "<div align=center>对不起，暂无留言！</div>";
    }else{
   //判断查询字符串 page 的值是否为空，如果为空则默认显示第一页
        if(!isset($_GET["page"]) || !is_numeric($_GET["page"])){
            $page=1;
        }else{
            $page=intval($_GET["page"]);
        }
         $pagesize=3;                             //规定每页显示 3 条留言
        if($total%$pagesize==0){                  //获取总页数
            $pagecount=intval($total/$pagesize);
        }else{
           $pagecount=ceil($total/$pagesize);
        }
   $sql=mysql_query("select * from tb_leaveword order by createtime
   desc limit ".($page-1)*$pagesize.",$pagesize   ",$conn);
         while($info=mysql_fetch_array($sql)){        //通过 while 循环显示所有留言

 ?>
…    //显示用户信息和用户留言信息
    <?php
          $sqlu=mysql_query("select usernc from tb_user where id='".$info["userid"]."'",$conn);
          $infou=mysql_fetch_array($sqlu);
 if($infou["usernc"]==$_SESSION["unc"]){  //判断登录用户是否为留言的发表者，如果是则显示“编辑”按钮
         ?>
…//显示“编辑”按钮
<?php
              }
```






```
        ?>
             
        <?php
                if($_SESSION["unc"]!=""){
            $sqld=mysql_query("select usertype from tb_user where usernc='".$_SESSION["unc"]."'",$conn);
              $infod=mysql_fetch_array($sqld);
            if($infod["usertype"]==1){    //判断当前登录用户是否为管理员，如果是则显示"删除"按钮
            ?>
            <a href="deleteleaveword.php?id=<?php echo $info[id];?>" class="a1">删除</a>
        <?php
                }
               }
        ?>
            …      //显示用户信息及用户留言信息
          <?php
                }
             }
            ?>
```

说明：

上述代码中，实现留言信息的分页显示是通过 MySQL 数据库的扩展关键字 limit 来实现的。该关键字后跟两个参数，其中第一个参数用于指定要显示记录的起始位置，而第二个参数用于指定所要显示的记录条数。

9.4.4 将留言信息进行分页显示

发表完留言后，可以通过查看留言页面查看所有的用户留言内容。由于用户的留言数目较多，如果在同一页面中显示所有留言信息，会给用户浏览带来很大的不便，所以通过分页的方式显示用户留言内容是个不错的选择。在实现用户留言内容分页显示时，主要应用 is_numeric()函数判断用户通过 GET 方法提交的数据是否为数值型，并通过 ceil()函数对页码数据进行向上取整。下面将对这两个函数进行介绍。

1. is_numeric()函数

如果该函数的参数为数字或数字字符串则返回 true，否则返回 false。其语法格式如下：

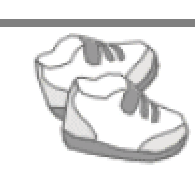
```
bool is_numeric ( mixed var )
```

其中，参数 var 为要进行判断的数据。

2. ceil()函数

ceil()函数用于对浮点数进行向上取整。其语法格式如下：

```
float ceil ( float value )
```





Note

其中，参数 value 为要进行向上取整的数据。

在显示用户留言时，根据传入当前页面的查询字符串 page 的值来决定所要显示的记录范围，创建分页超链接。关键代码如下：

```
<table width="550" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td width="351"><div align="left">共有留言 <?php echo $total;?> 条 
            每页显示<?php echo $pagesize;?> 条 第 <?php echo $page;?> 
            页/共 <?php echo $pagecount;?> 页</div>
        </td>
        <td width="199"><div align="right">
            <a href="<?php echo $_SERVER['PHP_SELF']?>?page=1" class="a1">首页</a> 
            <a href="<?php echo $_SERVER["PHP_SELF"]?>?page=
<?php
    if($page>1)
        echo $page-1;
    else
        echo 1;
?>
            " class="a1">上一页</a> 
            <a href="<?php echo $_SERVER["PHP_SELF"]?>?page=
<?php
    if($page<$pagecount)
        echo $page+1;
    else
        echo $pagecount;
?>
            " class="a1">下一页</a> 
<a href="<?php echo $_SERVER['PHP_SELF'];?>?page=<?php echo $pagecount;?>" class="a1">尾页</a></div>
        </td>
    </tr>
</table>
```

9.5 编辑留言

9.5.1 编辑留言概述

为了保证留言者的留言信息不被他人随意修改，用户只能对本人的留言内容进行修改，无权修改他人的留言内容。编辑留言页面的运行效果如图 9.14 所示。在该页面中单击“编辑”超链接，打开“编辑留言”窗口，修改留言内容后单击“编辑”按钮，即可完成修改留言操作。



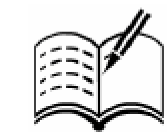




图 9.14　编辑留言页面

9.5.2　编辑留言页面设计

为了使用户能够查看到原留言的内容，在设计编辑窗口时，采用一个弹出窗口作为编辑表单。编辑留言页面的设计效果如图 9.15 所示。



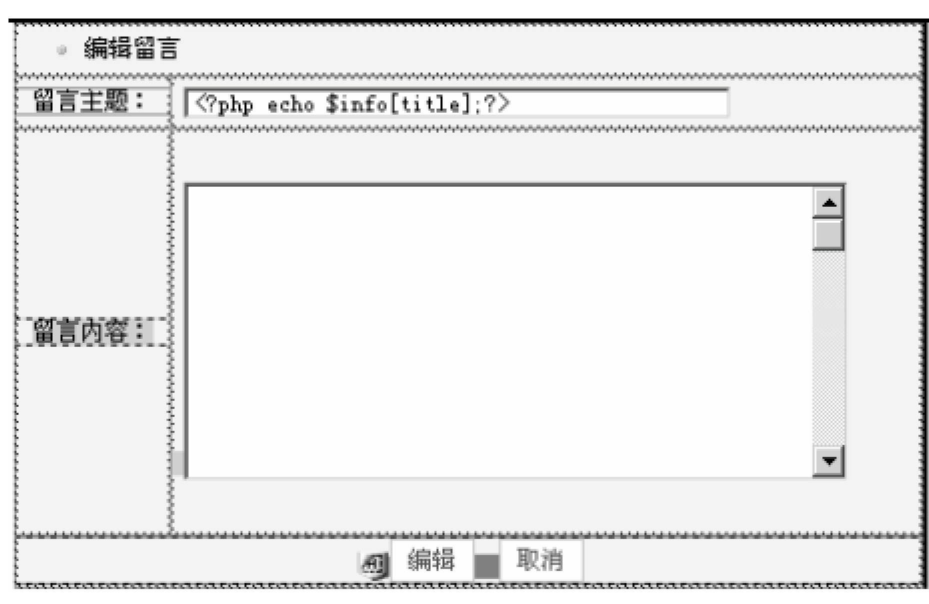
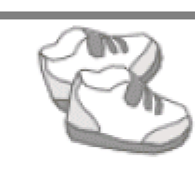


图 9.15　编辑留言页面的设计效果





编辑留言页面的设计流程如下：

（1）用户登录后，根据数据库中所保存的用户信息与用户的登录信息进行比较，如果二者相同则在其留言后显示“编辑”超链接。代码如下：

```
<?php
    $sqlu=mysql_query("select usernc from tb_user where id='".$info["userid"]."'",$conn);
    $infou=mysql_fetch_array($sqlu);
    if($infou["usernc"]==$_SESSION["unc"]){
?>
    <a href="javascript:openeditwindow(<?php echo $info[id];?>)" class="a1">编辑</a>
<?php
        }
?>
```

（2）当用户单击“编辑”超链接后，将通过自定义的 openeditwindow()方法打开一个新编辑窗口来编辑留言。代码如下：

```
<script language="javascript">
    function openeditwindow(x){            //定义 openeditwindow()方法打开弹出窗口
window.open("editleaveword.php?id="+x,"newframe","top=100,left=200,width=450,
height=280,menubar=no, location=no,scrollbars=no,status=no");
    }
</script>
```

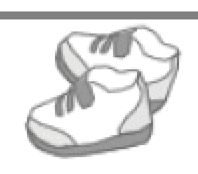
编辑留言页面涉及到的 HTML 表单中的重要元素如表 9.2 所示。

表 9.2　编辑留言页面涉及到的 HTML 表单中的重要元素

名　称	类　型	含　义	重 要 属 性
form1	form	表单	name="form1" method="post" action="<?php echo $_SERVER["PHP_SELF"];?>" onSubmit="return chkinput(this)"
title	text	发件人	name="title" type="text" class="inputcss" size="45" value="<?php echo $info[title];?>"
content	textarea	收件人	name="content" cols="52" rows="12" class="inputcss"
id	hidden	留言内容 ID	type="hidden" name="id" value="<?php echo $_GET[id];?>"
submit	submit	“编辑”按钮	type="submit" value="编辑" class="buttoncss" name="submit"
reset	reset	“取消”按钮	type="reset" name="reset" value="取消" class="buttoncss"

9.5.3　编辑留言内容功能实现

为了保证用户的留言内容不被他人私自更改，在具体实现该功能时，采用每个用户只能对自己的留言内容进行更改的方式。其实现原理如下：首先，将标识用户身份的 session 变量的值与数据库中该条留言对应的留言者进行比较，如果二者相同，则说明该条留言是由当前登录用户所发表的，在该条留言后显示“编辑”超链接。当用户单击“编辑”超链接后，即可实现留言信息的更改，并且在关闭弹出窗口的瞬间，留言信息的显示页面也会自动进行刷新。





执行编辑留言操作的 editleaveword.php 文件的代码如下：

```
<?php
include_once("conn.php");
$id=$_GET["id"];
//判断是否单击了“编辑”按钮，如果是则执行如下用于实现用户留言信息更改的代码，并刷新父窗口和关闭弹出窗口
if($_POST["submit"]!=""){
  if(mysql_query("update tb_leaveword set title='".$_POST["title"]."',content='".$_POST["content"]."'
where id='".$_POST["id"]."'",$conn)){
    echo "<script>alert('留言更改成功！');window.opener.location.reload();window.close();</script>";
     }else{
    echo "<script>alert('留言更改失败！');history.back();</script>";
  }
 exit;
}
$sql=mysql_query("select * from tb_leaveword where id='".$id."'",$conn);
     //如果用户未单击“编辑”按钮，则查询该用户的留言内容，并最终将未编辑的留言内容显示在表单中
$info=mysql_fetch_array($sql);
?>
```

说明：

编辑用户留言时，关闭弹出窗口、自动刷新父窗口，是通过如下语句实现的。

```
window.opener.location.reload();
```

上述代码的实现原理是：在用 window.close()语句关闭弹出窗口前，调用父窗口的 reload()方法实现父窗口的刷新。

9.6　删除留言

9.6.1　删除留言概述

如果用户发表不法留言，留言本管理员有权对其进行删除。管理员登录后将在每条留言后显示“删除”超链接，当管理员单击该超链接后，将弹出一个对话框，提示管理员是否真正删除该条留言。如果单击“确定”按钮则该条留言将被删除，反之不做任何操作。

说明：

本模块的管理员用户名为 mr，密码为 mrsoft。

删除留言页面的运行效果如图 9.16 所示。



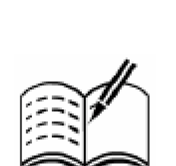




图 9.16　删除留言页面

9.6.2　删除留言页面设计

删除留言页面的设计效果如图 9.17 所示。

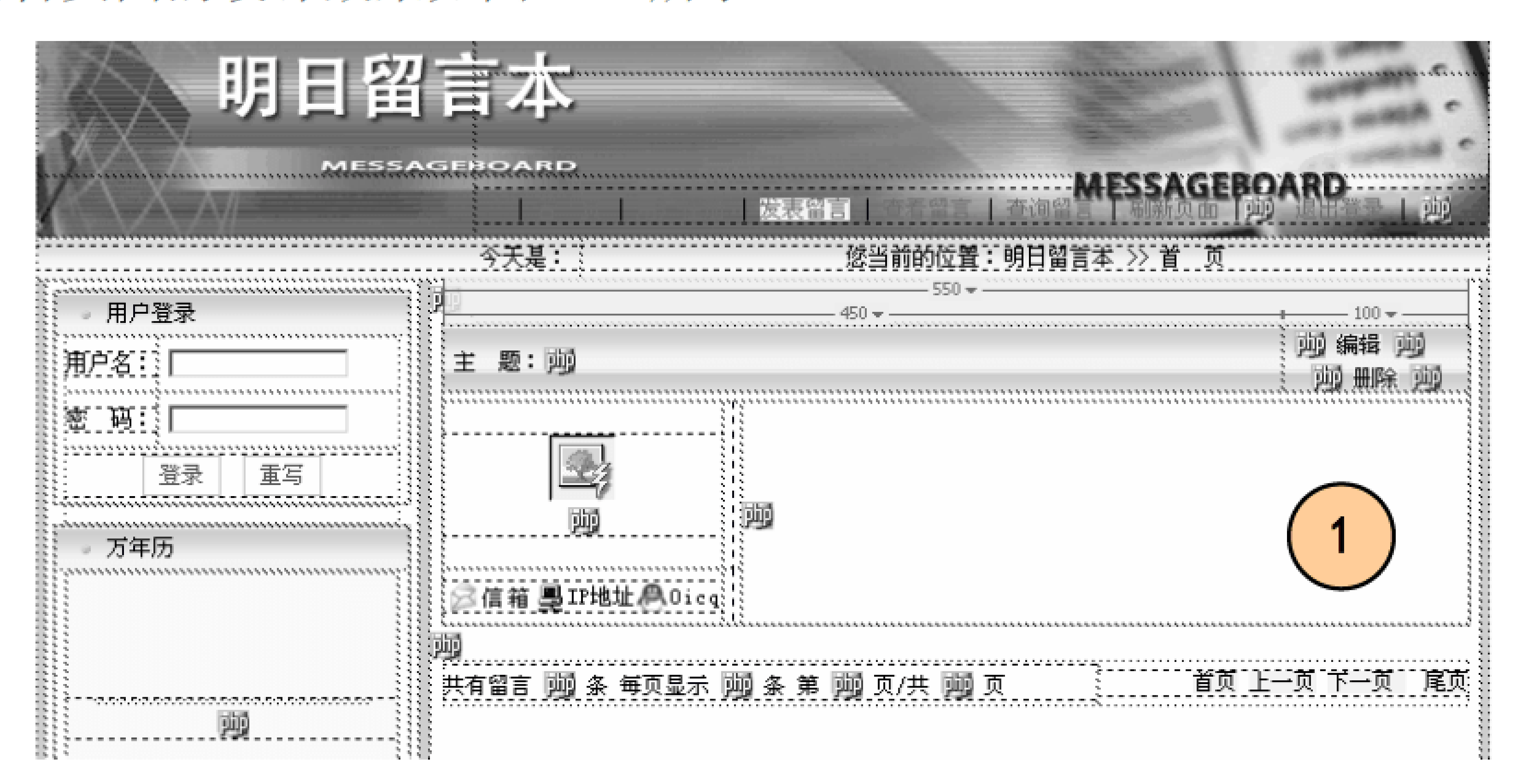


图 9.17　删除留言页面的设计效果

判断当前登录用户是否为管理员，如果是则显示“删除”超链接。代码如下：

```
<?php
    if($_SESSION["unc"]!=""){                    //判断用户是否登录
        $sqld=mysql_query("select usertype from tb_user where usernc='".$_SESSION["unc"]."'",$conn);
```






```
            $infod=mysql_fetch_array($sqld);
                if($infod["usertype"]==1){                  //判断登录用户是否为管理员
    ?>
        <a href="javascript:if(window.confirm('确定删除该留言信息么？')==true)
{window.location.href='deleteleaveword.php?id=<?php echo $info[id];?>';}" class="a1">删除</a> //如果是则显示“删除”超链接
        <?php
                }
                }
        ?>
```

9.6.3　删除留言内容功能实现

当管理员确认删除留言后，将进入留言删除处理页 deleteleaveword.php 中，根据超链接传递的 ID 值，执行删除操作。deleteleaveword.php 的代码如下：

```
<?php
include_once("conn.php");                           //包含数据库连接文件
if(mysql_query("delete from tb_leaveword where id='".$_GET["id"]."'",$conn)){
   echo "<script>alert('留言删除成功！');history.back();</script>";
}else{                                              //删除指定的留言，并给出提示
   echo "<script>alert('留言删除失败！');history.back();</script>";
}
?>
```

9.7　查　询　留　言

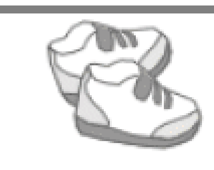
9.7.1　查询留言概述

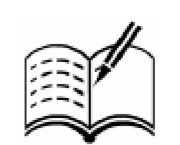
随着留言本使用年限的增多，留言的数量会越来越多。为了方便浏览者的浏览和快速定位到指定的留言，在制作留言本时，设计了可以按留言主题和留言内容进行模糊查找、按留言者进行匹配查找的查询留言功能。

查询留言页面的运行效果如图 9.18 所示。



图 9.18　查询留言页面





Note

9.7.2 查询留言页面设计

查询留言页面的设计效果如图 9.19 所示。

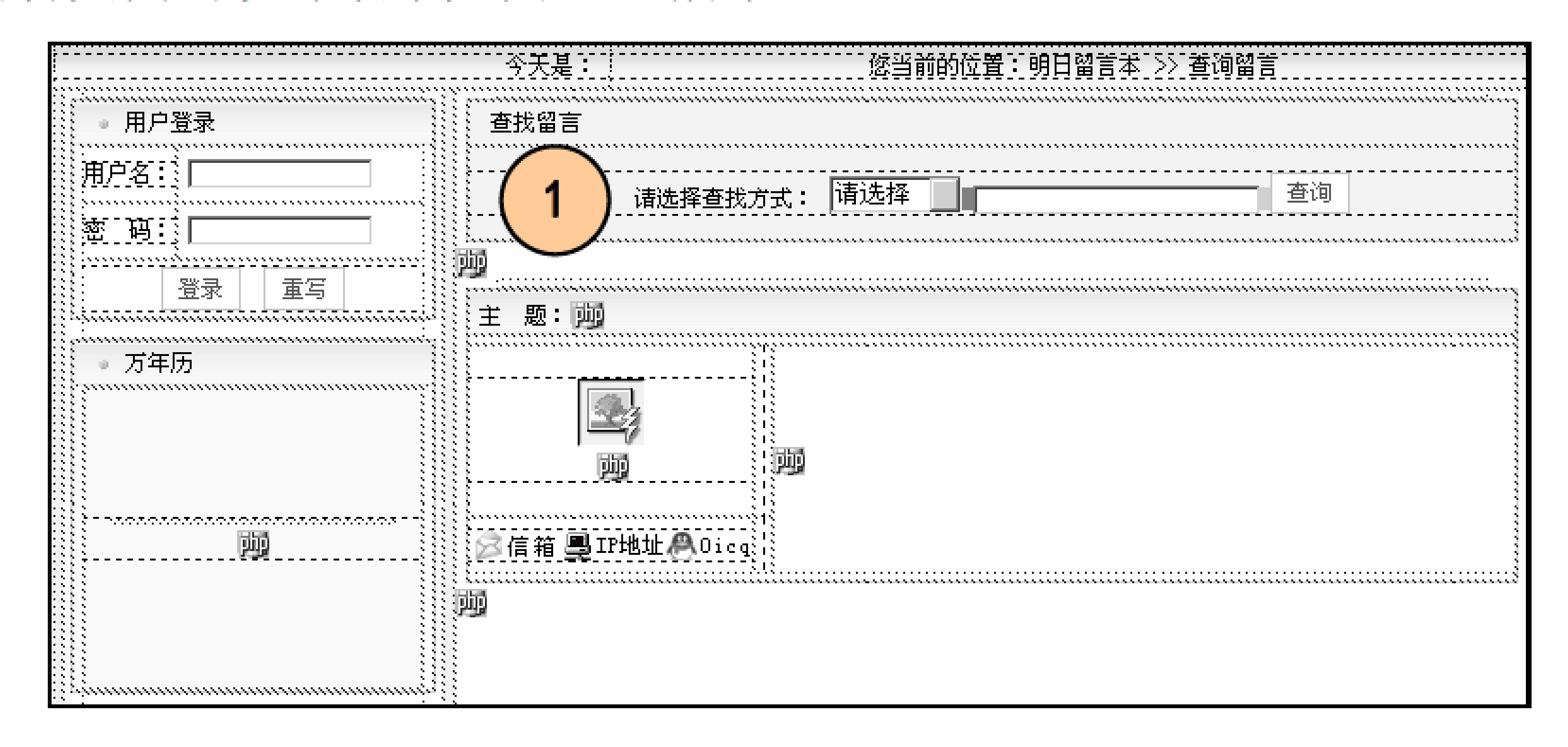


图 9.19 查询留言页面的设计效果

为了方便用户，设计了可以按多种方式进行查找的查询表单。代码如下：

```
<div align="center">请选择查找方式：
  <select name="type">
       <option value="">请选择</option>
     <option value="1">主题</option>
     <option value="2">内容</option>
     <option value="3">留言者</option>
  </select> <input type="text" name="keyword" size="25" class="inputcss"> 
<input type="submit" value="查询" class="buttoncss" name="submit">
</div>
```

查询留言页面涉及到的 HTML 表单中的重要元素如表 9.3 所示。

表 9.3 查询留言页面涉及到的 HTML 表单中的重要元素

名 称	类 型	含 义	重 要 属 性
form1	form	表单	name="form1" method="post" action="<?php echo $_SERVER["PHP_SELF"];?> " onsubmit="return chkinput_search(this)"
type	select	查询类型	name="type"
keyword	text	查询关键字	type="text" name="keyword" size="25" class="inputcss"
submit	submit	“查询”按钮	type="submit" value="查询" class="buttoncss" name="submit"

9.7.3 查询留言内容

当用户按要求输入了查找关键字，选择了查找方式后，单击“查询”按钮，数据将被提交到 searchword.php 页，在该页完成数据的查询操作，并且输出查询结果。searchword.php 的关键代码如下：

```
<?php
```






```
if($_POST["submit"]!=""){               //判断用户是否单击了“查询”按钮，如果是则开始查询工作
 $type=$_POST["type"];
 $keyword=$_POST["keyword"];
  if($type==1){                         //根据用户提交的查询类型，定义查询语句
     $sql=mysql_query("select * from tb_leaveword where title like '%".$keyword."%'",$conn);
  }elseif($type==2){
     $sql=mysql_query("select * from tb_leaveword where content like '%".$keyword."%'",$conn);
  }elseif($type==3){
     $sql0=mysql_query("select id from tb_user where usernc='".$keyword."'",$conn);
     $info0=mysql_fetch_array($sql0);
     $sql=mysql_query("select * from tb_leaveword where userid='".$info0["id"]."'",$conn);
  }
  $info=mysql_fetch_array($sql);
  if($info==false){           //判断是否查找到相关内容，如果没查找到则显示提示信息
     echo "<br><br><div align=center>对不起，没有查找到您要查找的内容！</div>";
  }else{
      do{                                //通过循环显示查询结果
?>
  …//显示查找内容
<?php
      }while($info=mysql_fetch_array($sql));
       }
  }
?>
```

9.8 技 术 提 炼

9.8.1 将数据保存到数据库中

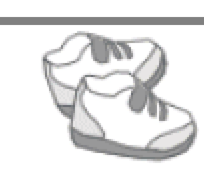
将用户发表的留言保存到 MySQL 数据库是通过 SQL 语言的 insert 语句实现的。PHP 通过 mysql_query()函数向 MySQL 数据库中发送 SQL 命令，由此可见，该函数在数据库编程过程中具有极其重要的作用。

mysql_query()函数：用于向指定的连接标识符所关联的数据库服务器中发送一条查询语句。其语法格式如下：

```
resource mysql_query (string query [, resource link_identifier])
```

参数说明如下。

- ☑ query：必选参数，用于指定向数据库中发送的查询语句。
- ☑ link_identifier：可选参数，代表 mysql_connect()函数成功连接数据库后所返回的连接标识；如果省略了该参数，则使用最近一次与数据库建立连接后所返回的连接标识。



Note

9.8.2　通过 JavaScript 实现弹出窗口

通过 JavaScript 实现弹出窗口，主要是通过调用 window 对象的 open()方法实现的，并将实现调用的代码封装到一个单独的自定义方法中，最后通过 HTML 元素的 onclick()事件对该方法进行调用。

open()方法的语法格式如下：

```
open("url","name","features")
```

该方法的参数说明如表 9.4 所示。

表 9.4　open()方法的参数说明

参数取值	说　明
url	可选参数。用于指定弹出新窗口的 URL 地址，如果省略该参数或其值是一个空字符串，那么新窗口就不显示任何文档
name	新窗口的名字。如果这个名字是一个已经存在的窗口，那么 open()方法就不再创建一个新窗口，而只是返回这个窗口的引用。在这种情况下，参数 features 将被忽略
features	主要是显示窗口的一些特征。该参数如果不设定，则显示这个窗口的所有特征。features 的常用特征参见表 9.5

表 9.5　参数 features 的常用特征

名　称	用　法
channelmode	指定窗口是否应该以频道的方式显示，取值有（yes/no 或 1/0）
fullscreen	指定窗口是否全屏显示，取值有（yes/no 或 1/0）
menubar	指定窗口是否有菜单栏，取值有（yes/no 或 1/0）
scrollbars	指定窗口是否有水平及竖直滚动条，取值有（yes/no 或 1/0）
status	指定窗口是否有状态栏，取值有（yes/no 或 1/0）
toolbar	指定窗口是否有工具栏，取值有（yes/no 或 1/0）
location	指定窗口是否有地址栏，取值有（yes/no 或 1/0）
directions	指定新建窗口是否有标准目录按扭，取值有（yes/no 或 1/0）
resizable	指定新窗口大小是否可以调整，取值有（yes/no 或 1/0）
top	以像素为单位指定新窗口上沿与屏幕上沿的距离
left	以像素为单位指定新窗口左沿与屏幕左沿的距离
width	以像素为单位指定新窗口的宽度
height	以像素为单位指定新窗口的高度

9.8.3　包含文件函数

在开发 Web 程序时，经常会在不同的页面中实现相同的功能或执行相同的代码。为了提高程序开发效率，可以将相同的代码存储到单独的文件中，然后用包含语句在页面中包含该文件即可。

include_once 语句：用于在脚本执行期间包含并运行指定的文件。其语法格式如下：

```
include_once(filename)
```



其中，参数 filename 用于指定要包含的文件。

9.8.4 MySQL 数据库的函数



PHP 中提供了一组操作 MySQL 数据库的函数，利用这些函数可以方便地对 MySQL 数据库进行管理。

mysql_fetch_array()函数：获得数据库中满足函数 mysql_query()中的 SQL 语句的记录。其返回值是一个数组，该数组的下标可以是字段名，也可以是索引下标，数组元素的值是某个字段的内容。该函数会使记录指针自动向下移动，当移动到最后一行将返回一个 false 值。其语法格式如下：

```
array mysql_fetch_array(int result,int[result_type])
```

其中参数说明如表 9.6 所示。

表 9.6 mysql_fetch_array()函数参数说明

参数取值	说明
result	必选参数。mysql_query()函数向 MySQL 数据库发送查询命令的返回标识
result_type	可选参数。如果该参数取 MYSQL_BOTH，则 mysql_fetch_array()函数将返回一个同时包含关联索引和数字索引的数组；如果该参数取 MYSQL_ASSOC，则返回一个关联索引的数组；如果该参数取 MYSQL_NUM，则返回一个数字索引的数组

说明：

与该函数功能类似的函数还有 mysql_fetch_rows()、mysql_result()、mysql_fetch_object()等。mysql_fetch_rows()函数返回的数组的下标为数值索引下标。函数 mysql_result()有两个参数，其中第一个参数也是 mysql_query()的返回结果，而第二个参数可以是字段的偏移量或者字段名。一定要注意，它返回的结果不是数组，而是 MySQL 结果集中一个单元的内容。函数 mysql_fetch_object()的返回结果是个对象，使用时只能通过字段名来返回结果。

第10章

在线论坛模块

（自学视频、源程序：配套资源\mr\10\）

随着网络的飞速发展，人们对网站开发的要求越来越高。从最初简单的静态页面，到今天日益普及的动态网站，无论是其功能结构、组成元素，还是设计方法、实现过程等，均发生了质的飞跃。如今评价一个网站开发得是否成功的标准更高，不仅要看设计的界面是否美观，还要看网站的功能是否齐全。这其中便包括一项必不可少的功能——在线论坛。目前，在线论坛在商业网站中所扮演的角色极为重要，直接关系到企业与用户、用户与用户之间能否畅通、便捷地进行沟通与交流。因此，能否开发一个完整而美观的在线论坛便成为程序员的必修课程。在本章就对这门必修课程进行详细的讲解，以实战的形式全景展示一个在线论坛的开发过程。

- 分页技术
- 网页访问权限的设置
- 搜索引擎的设计
- 页面刷新技术



10.1　在线论坛概述



10.1.1　模块概述

论坛是一个完整网站必不可少的功能之一。通过在线论坛，用户不仅可以阐述自己的观点，而且还可以回复别人的文章，从而达到相互学习和交流的目的。

在线论坛主要包括以下功能：查看帖子、发表帖子、回复帖子、用户注册登录、管理员登录、添加主题分类、删除主题和回复信息，以及注销。其主页运行效果如图 10.1 所示。



图 10.1　在线论坛主页

10.1.2　功能结构

本论坛主要实现以下功能：一方面是供用户使用的功能，包括用户的注册、登录，查看专区类别，查看专区的帖子主题，查看主题帖子的内容，发布主题帖子和回复帖子；另一方面是供管理员使用的功能，包括管理员的登录、栏目管理、帖子主题的管理、回复帖子的管理，以及注册会员的管理。在线论坛的功能结构如图 10.2 所示。

10.1.3　权限分配

在本论坛系统中，对于不同的用户给予了不同的权限。没有注册的用户，可以直接访问本论坛，浏览论坛的专区帖子，但是不能回复和发布主题帖子；注册的用户既可以浏览论坛的所有帖子，又能够对帖子进行回复和发布帖子；论坛管理员既可以对论坛的专区、帖子进行管理，也可以对论坛的注册用户进行管理。在线论坛的权限分配如图 10.3 所示。







Note

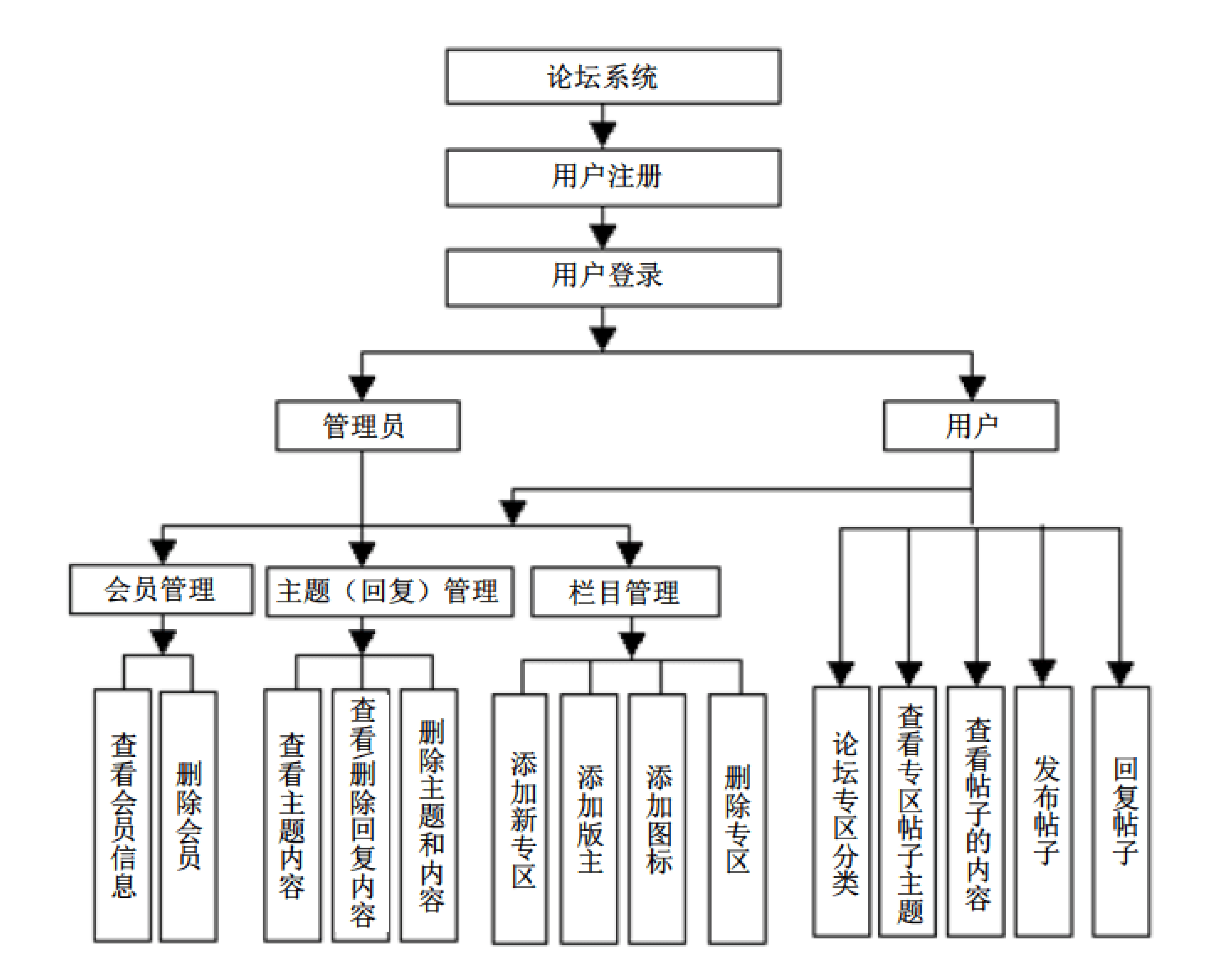


图 10.2　在线论坛功能结构图

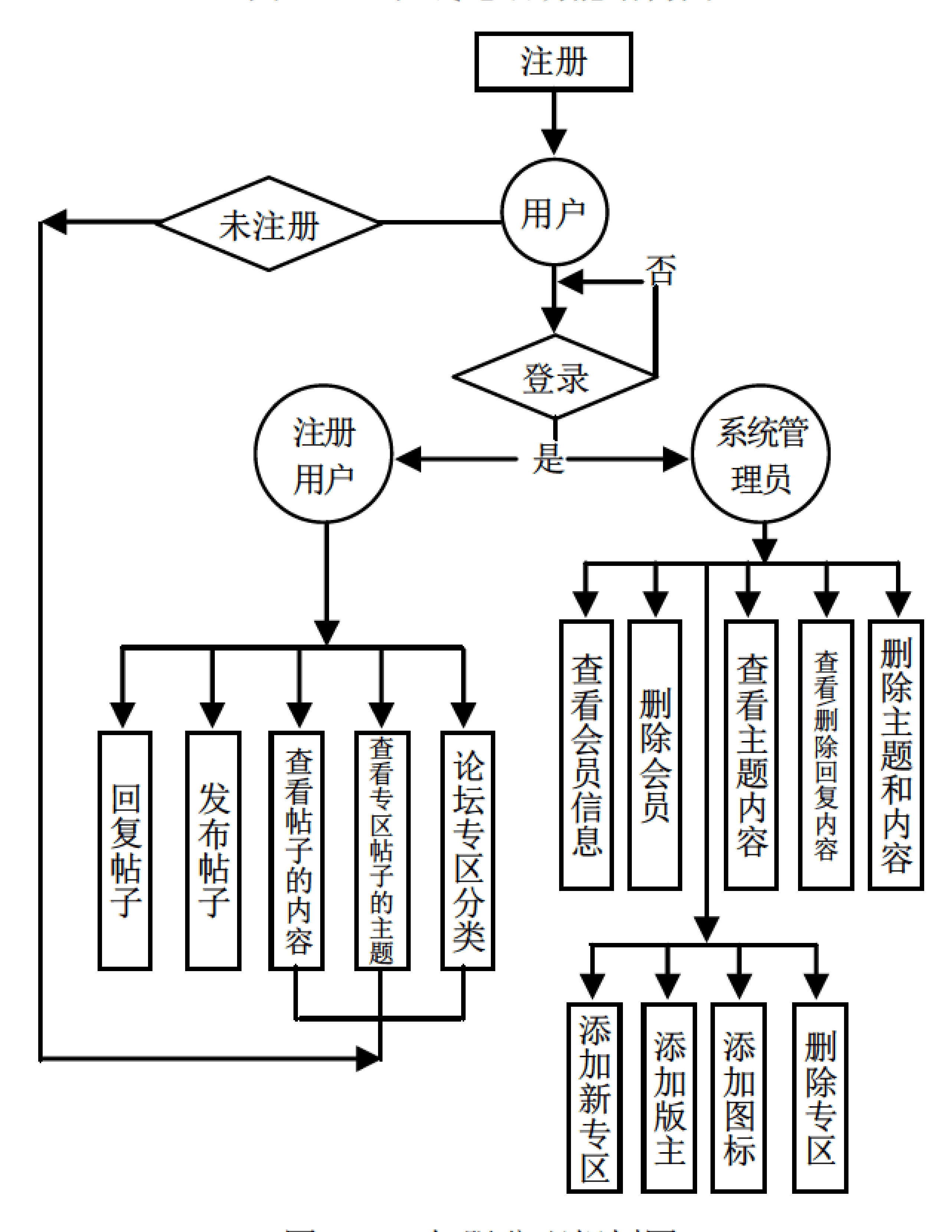


图 10.3　权限分配规划图

10.1.4　程序预览

在线论坛是由多个功能模块组成的，下面列出几个典型功能模块的运行效果图，让读



者对本系统有一个初步的了解和认识。

在线论坛主页的运行效果如图 10.4 所示，其主要功能是展示论坛中的帖子专区，以及每个专区中包含帖子的总数量和当前发布帖子的数量。



图 10.4　在线论坛主页的运行效果

分页浏览在线论坛指定专区的帖子，其中包括主题、作者、回复次数、访问次数以及发布时间，其运行效果如图 10.5 所示。

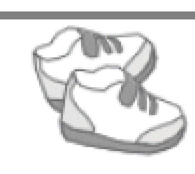


图 10.5　浏览论坛中帖子的主题

查看论坛中帖子的具体内容，包括帖子主题的内容，以及对应的回复内容，并且对回复内容进行分页输出。其运行效果如图 10.6 所示。



图 10.6　浏览论坛中帖子内容



发布帖子的页面运行效果如图 10.7 所示。



图 10.7　发布帖子

10.2　数据库设计

在线论坛主要是通过数据库的支持来存储大量的数据，因此数据库设计的成败是该系统能否成功运行的关键。前面已经对该程序要实现的功能进行了详细的分析，本节将针对前面提及的功能设计一个合理的数据库。

10.2.1　数据库设计

针对在线论坛系统的功能结构，创建一个名为 mr_mysql 的数据库，其中包含的数据表如图 10.8 所示。



服务器: localhost ▸ 数据库: mr_mysql

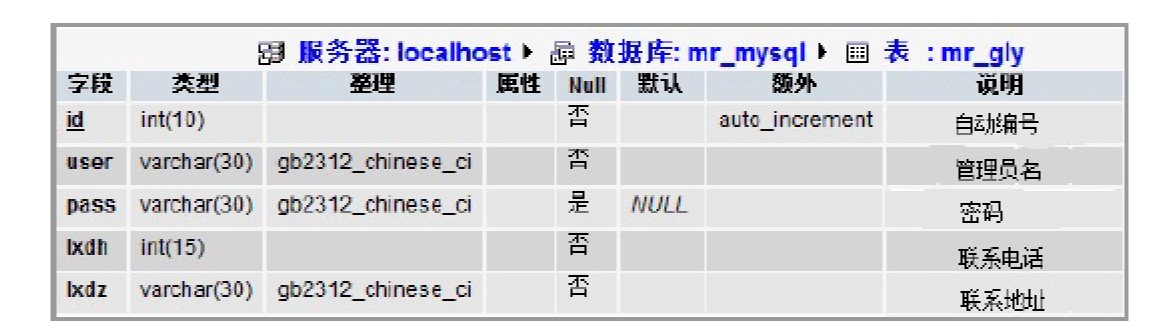
表	操作	记录数	类型	整理	说明
mr_gly		1	MyISAM	gb2312_chinese_ci	管理员的注册信息
mr_hflb		3	MyISAM	gb2312_chinese_ci	存储回复帖子内容
mr_lttb		32	MyISAM	latin1_swedish_ci	存储论坛中使用的表情图标
mr_user		1	MyISAM	gb2312_chinese_ci	存储用户的注册信息
mr_zqfl		5	MyISAM	gb2312_chinese_ci	存储论坛包括的专区类型
mr_zqlb		5	MyISAM	gb2312_chinese_ci	存储发布帖子的内容
6 个表	总计	47	MyISAM	latin1_swedish_ci	

图 10.8　数据表相关信息

10.2.2　数据表设计

了解了数据库的基本结构，下面详细介绍数据库中各个表的结构。

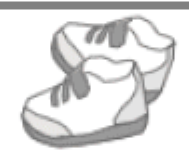
mr_gly 表用于存储管理员注册信息，其结构如图 10.9 所示。



服务器: localhost ▸ 数据库: mr_mysql ▸ 表 : mr_gly

字段	类型	整理	属性	Null	默认	额外	说明
id	int(10)			否		auto_increment	自动编号
user	varchar(30)	gb2312_chinese_ci		否			管理员名
pass	varchar(30)	gb2312_chinese_ci		是	NULL		密码
lxdh	int(15)			否			联系电话
lxdz	varchar(30)	gb2312_chinese_ci		否			联系地址

图 10.9　mr_gly 表



mr_user 表用于记录用户注册的相关信息，其结构如图 10.10 所示。

服务器: localhost ▸ 数据库: mr_mysql ▸ 表 : mr_user

字段	类型	整理	属性	Null	默认	额外	说明
id	int(10)			否		auto_increment	自动编号
username	varchar(30)	gb2312_chinese_ci		否			用户名
zsxm	varchar(30)	gb2312_chinese_ci		否			密码
password	varchar(30)	gb2312_chinese_ci		否			真实姓名
sex	varchar(30)	gb2312_chinese_ci		否			性别
shengri	date			否			用户的生日
lxdh	int(15)			否			联系电话
qq	int(20)			否			QQ号码
tx	varchar(30)	gb2312_chinese_ci		是	NULL		注册时选择的头像
email	varchar(30)	gb2312_chinese_ci		否			用户的邮箱地址
grzy	varchar(30)	gb2312_chinese_ci		否			用户的个人主页
lxdz	varchar(50)	gb2312_chinese_ci		否			用户的联系地址



图 10.10　mr_user 表

mr_hflb 表用于记录用户在回复帖子时的信息，包括帖子的主题、内容、回复时间等，其结构如图 10.11 所示。



服务器: localhost ▸ 数据库: mr_mysql ▸ 表 : mr_hflb

字段	类型	整理	属性	Null	默认	额外	说明
id	int(10)			否		auto_increment	自动编号
hfzt	varchar(50)	gb2312_chinese_ci		否			回复帖子的主题
hfnr	mediumtext	gb2312_chinese_ci		否			回复的内容
hfsj	date			否			回复的时间
username	varchar(30)	gb2312_chinese_ci		否			回复的用户名
ljid	int(10)			否			连接的id的值
zq	varchar(30)	gb2312_chinese_ci		否			它属于哪个专区

图 10.11　mr_hflb 表

mr_lttb 表的作用是记录论坛中使用的表情图标，其结构如图 10.12 所示。

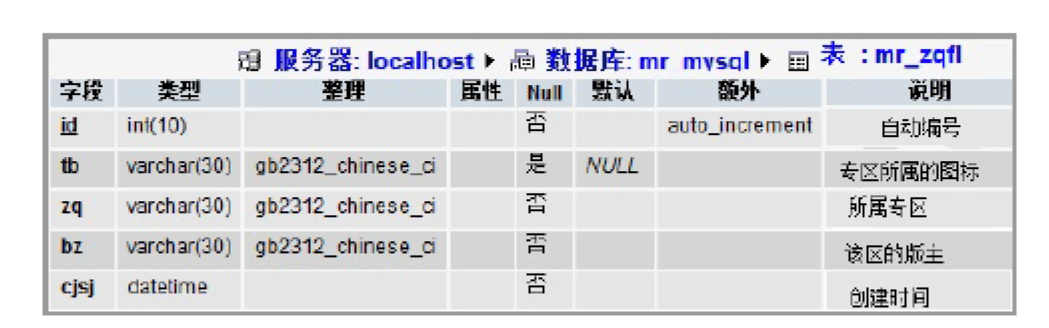


服务器: localhost ▸ 数据库: mr_mysql ▸ 表 : mr_lttb

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
xsbs	varchar(10)	latin1_swedish_ci		否			
xq	longblob		BINARY	是	NULL		心情图片

图 10.12　mr_lttb 表

mr_zqfl 表记录了在论坛中有几个专区，及每个专区的详细信息，包括专区的图标、版主、创建的时间，其结构如图 10.13 所示。



服务器: localhost ▸ 数据库: mr_mysql ▸ 表 : mr_zqfl

字段	类型	整理	属性	Null	默认	额外	说明
id	int(10)			否		auto_increment	自动编号
tb	varchar(30)	gb2312_chinese_ci		是	NULL		专区所属的图标
zq	varchar(30)	gb2312_chinese_ci		否			所属专区
bz	varchar(30)	gb2312_chinese_ci		否			该区的版主
cjsj	datetime			否			创建时间

图 10.13　mr_zqfl 表

mr_zqlb 表记录的是在发布主题中发布的帖子的详细信息，包括帖子的主题、内容、发布时间和访问记录等，其结构如图 10.14 所示。

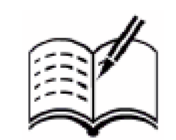


服务器: localhost ▸ 数据库: mr_mysql ▸ 表 : mr_zqlb

字段	类型	整理	属性	Null	默认	额外	说明
id	int(10)			否		auto_increment	自动编号
zq	varchar(30)	gb2312_chinese_ci		否			发布帖子所属专区
xq	varchar(30)	gb2312_chinese_ci		是	NULL		表情图标
zhuti	varchar(50)	gb2312_chinese_ci		否			帖子的主题
neirong	mediumtext	gb2312_chinese_ci		否			帖子的内容
username	varchar(30)	gb2312_chinese_ci		否			发布帖子的用户名
fbsj	date			否			帖子发布的时间
fwjl	int(30)			否			发布帖子的记录数

图 10.14　mr_zqlb 表





Note

到此，对在线论坛系统的数据库设计已经介绍完毕，接下来的工作就是程序的设计。

10.2.3　连接数据库

在对数据库进行操作前，首先要建立与数据库的连接，即使用 config.php 文件进行数据库连接。首先使用 mysql_connect 语句连接本地服务器 loacalhost，用户名是 root，密码是 111；然后使用 mysql_select_db 语句选择数据库 mr_mysql；最后，通过 mysql_query() 函数设置数据库的编码格式为 gb2312。其详细代码如下：

```
<!--*******************************Config.php*********************************-->
<?php
$id=mysql_connect("localhost","root","111");                //连接数据库服务器
mysql_select_db("mr_mysql",$id);                            //选择数据库
mysql_query("set names gb2312");                            //设置编码格式
?>
```

如果某个页面中需要进行数据库的操作，在页面的前台直接包含该文件即可。例如：

```
<?php
include("config.php");                                      //包含数据库连接文件
?>
```

10.3　用 户 注 册

10.3.1　用户注册概述

在用户注册时，需要用户填写的必填信息包括用户名、密码和 E-mail 等都用“*”标注出来，只有这些必填信息添加完整并正确时，才能进行注册。在用户名填写完成后，用户可以通过单击“检测用户名”超链接，检测所填写的用户名是否被注册，如果所填写的用户名已经注册过，则要填写其他的用户名来注册。注册页面的运行效果如图 10.15 所示。

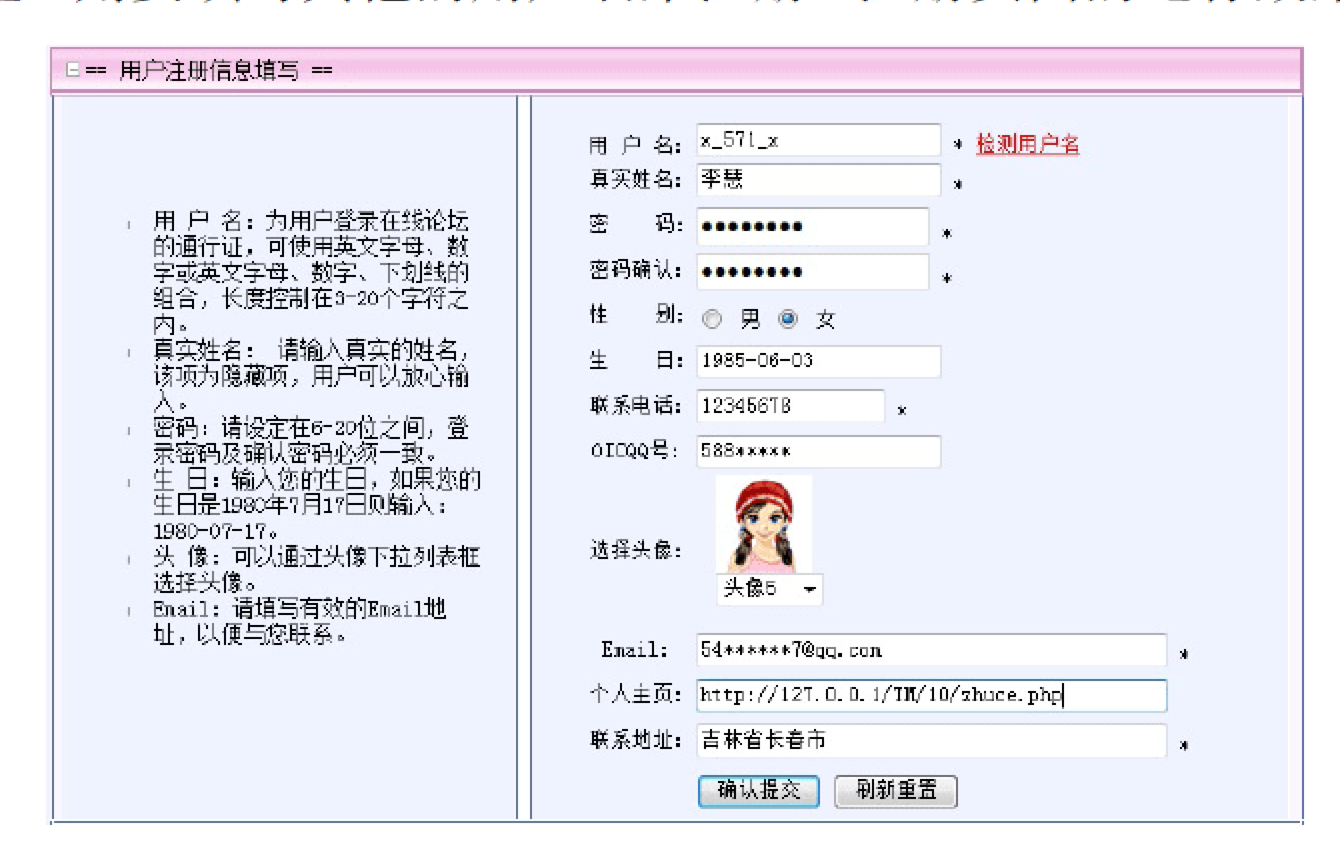
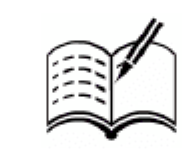


图 10.15　注册页面的运行效果





10.3.2　检测用户名是否被占用

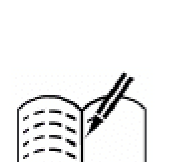
在注册过程中，用户名必须是唯一的，不能同时存在相同的用户名，因此首先要对输入的用户名进行检测。输入用户名，单击“检测用户名”超链接，将调用 Javascript 脚本，将输入的用户名提交到 submit_checkuser.php 文件中进行检测。submit_checkuser.php 文件的代码如下：

```
<?php
include "config.php";                                        //包含数据库连接文件
if(isset($_POST['zc_username']) && $_POST['zc_username']!=""){//判断提交的值是否存在
        $query="select * from mr_user where username='".$_POST['zc_username']."'";  // 定 义查询语句
        $result=mysql_query($query);                          //执行查询语句
        $nn=mysql_num_rows($result);                          //获取查询结果
        if ($nn==1) {                                         //判断用户名是否被占用
                echo"<script>alert('用户名已经存在，不能注册');history.back();</script>";
        } else {
                echo"<script>alert('恭喜您，可以用此用户名进行注册');history.back();</script>";
        }
}
?>
```

10.3.3　用户注册的实现

完成注册信息的填写后，通过 JavaScript 来判断用户填写的信息是否符合要求（如用户名使用什么字符、不可以超过多少个字节、日期的格式等），如果填写的用户信息正确，则单击“确认提交”按钮，将数据提交到数据处理页 zc_ok.php 中，最终将数据添加到指定的数据表中，完成用户注册的操作，并且将注册的用户名和密码赋给 SESSION 变量。数据处理文件 zc_ok.php 的代码如下：

```
<?php
session_start();                                              //初始化 SESSION 变量
include "config.php";                                         //包含数据库文件
$furl=getenv("HTTP_REFERER");                                 //获取 IP
if(isset($_POST['qrtj'])){                                    //对提交按钮进行判断
        $username=$_POST['zc_username'];
         $password=$_POST['zc_password'];
        $zsxm=$_POST['zsxm'];
         $sex=$_POST['sex'];
         $shengri=$_POST['shengri'];
         $lxdh=$_POST['lxdh'];
         $qq=$_POST['qq'];
         $tp=$_POST['ICO'];
         $email=$_POST['email'];
         $grzy=$_POST['grzy'];
         $lxdz=$_POST['lxdz'];
         $query="select * from mr_user where username='$username'";//在数据库中查询提交
```


Note

```
的用户名
            $result=mysql_query($query);
            if(mysql_num_rows($result)>0){                //对查询的记录数进行判断
                echo $username."已经被注册!</font>";      //若用户名注册过，则给出提示
           echo "<meta http-equiv=\"Refresh\" content=\"3;url=$furl\">3 秒钟转入注册页,请稍等...";    //跳转回注册页
            }else{
                $passwords=md5($password);
                $tp="image/tx/".$_POST['ICO'];
               $query="insert into mr_user (username,zsxm,password,sex,shengri,lxdh,qq,tx,email,grzy,lxdz)
    values('$username','$zsxm','$passwords','$sex','$shengri','$lxdh','$qq','$tp','$email','$grzy','$lxdz')";
                if(mysql_query($query)){
                    $_SESSION['username']=$username;    //通过 SESSION 获取提交的用户名
                    $_SESSION['password']=$password;    //通过 SESSION 获取提交的密码
                    echo "<font class=\"red\">您注册的信息如下！</font><br>";
                echo "<li class=\"huise03\">用户名:<font color=red>".$username."<br>";       //输出注册的用户名
                    echo "<li class=\"huise03\">E-Mail:<font color=red>".$email."<br>";       //输出注册的邮箱
                    echo "<li class=\"huise03\"><font color=red>".$username."</font>恭喜您注册成功！";
                    echo "<meta http-equiv=\"Refresh\" content=\"3;url=index.php\">5 秒钟转入主页,请稍等...";
                }else{
                    echo "<font class='#ff0000'>注册失败!!!</font>";
                    echo mysql_error();
                }
            }
    }

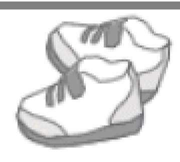
    ?>
```

完成用户注册后，就可以登录本论坛发布和回复帖子了。用户登录模块的实现比较简单，此处不再赘述，具体的源代码可以参见配套资源（文件名是 dljm.php 和 dl_ok.php）。

10.4 在线论坛前台

10.4.1 帖子浏览功能概述

笔者设计的这个在线论坛系统，其中的帖子是公开的，只要游客浏览本网站，就可以看到论坛中所有的帖子信息。因此，本论坛设计的第一个功能模块就是查看帖子的信息。它通过 index.php、lmzy.php 和 zthf.php 3 个文件来实现，分别可以浏览到论坛中都有什么专区、每个专区中都有什么主题的帖子、每个主题帖子的内容以及相关的回复信息。论坛主页 index.php 的运行效果如图 10.16 所示。



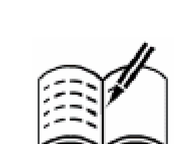




图 10.16　在线论坛主页运行效果

10.4.2　分页显示回复信息

该模块虽然是通过 3 个文件来实现的，但是使用的方式是相同的，下面以帖子详细信息浏览为例，讲解其具体的实现方法。zthf.php 页面如图 10.17 所示。

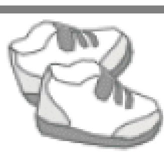


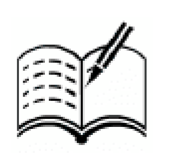
图 10.17　查看帖子内容

帖子详细内容和回复内容分页浏览功能的设计思路如下：首先连接到数据库，然后根据条件从数据库中查询数据，最后通过 while 语句循环输出查询结果。

在 zthf.php 文件中，首先根据超链接传递的 ID 值，从数据库中查询出指定帖子的信息，并且输出帖子的主题、发布人、帖子内容和发布时间。其关键代码如下：

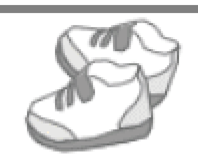
```
<?php
session_start();                                        //初始化 SEESION 变量
include("config.php");                                  //包含数据库文件
$page=(isset($_GET['page']))?$_GET['page']:"1";         //使用三元运算符对变量 page 进行判断
$zhuti=(isset($_GET['zhuti']))?$_GET['zhuti']:"PHP";//使用三元运算符对提交的主题进行判断
?>
```



Note

```
        /*........省略部分表单元素......*/
        <td width="776" height="30">    主题:『<?php echo $zhuti;?>』</td>
        /*........省略部分表单元素......*/
    <?php
$query="select * from mr_zqlb where id='".$_GET['recid']."'";//根据传递的ID值，在mr_zqlb表中查询相关数据
    $result=mysql_query($query);
    $ztrow=mysql_fetch_array($result);
    ?>
    <tr>
    <?php
$update="update mr_zqlb set fwjl=fwjl+1 where id='".$_GET['recid']."'"; //根据传递的ID值，更新数据表中对原帖的访问量
    $result=mysql_query($update);
    $query="select * from mr_user where username='".$ztrow['username']."'";          //查询发帖人的相关信息
    $result=mysql_query($query);
    $row=mysql_fetch_array($result);
    ?>
    <table width="136" height="145" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td height="24" align="center" valign="middle" class="style1"><?Php echo $ztrow['username'];?></td>
        </tr>
        <tr>
            <td height="64" align="center" valign="middle">
    <img src="<?php echo $row['tx'];?>"width="60" height="60"></td>
        </tr>
        <tr>
            <td height="26" align="center" valign="middle" class="style1">我是:<?php echo $row['sex'];?>生</td>
         </tr>
       <tr>
             <td height="30" align="center" valign="middle">
    <img src="images.php?recid=<?php echo $ztrow['xq'];?>"></td>
      </tr>
    </table>
    <td width="643" align="left" valign="top">
    <table width="100%"  border="0" align="center" cellpadding="0" cellspacing="0">
         <tr>
            <td width="80%" height="24" class="style1"> Email:<?php echo $row['email'];?> 
    QQ:<?php echo $row['qq'];?>   发表时间:<?php echo $ztrow['fbsj'];?></td>
            <td width="20%" class="style1">
    <a href="hftj.php?zhuti=<?php echo urlencode($zhuti);?>&recid=<?php echo $_GET['recid'];?>">回复</a></td>
         </tr>
         <tr>
           <td width="505" class="style2"> <?php echo $ztrow['neirong'];?> </td>
```



```
        </tr>
            </table>
```

然后，根据帖子的 ID 值从回复帖子数据表中查询出其对应的回复帖子，并且对查询结果进行分页输出。其关键代码如下：

```
<?php
$page_size=2;
$query="select * from mr_hflb where ljid='".$_GET['recid']."'";//根据传递的 ID 值，查询 mr_hflb 表中相关的回复
$result=mysql_query($query);
$message_count=mysql_num_rows($result);                    //获取回复的记录数
if($message_count>0){                                       //如果获取的记录数大于 0，则分页输出回复内容
?>
<table width="776" border="0" align="center" cellpadding="0" cellspacing="0">
<?php
$page_count=ceil($message_count/$page_size);               //对所分的页数取整
$offset=($page-1)*$page_size;                              //计算下一页从第几条数据开始循环
$quer="select * from mr_hflb where ljid='".$_GET['recid']."' order by id desc limit $offset ,$page_size";
$resul=mysql_query($quer);
while($myrow=mysql_fetch_array($resul)){
?>
  <tr>
    <td height="28" colspan="3" valign="middle" background="image/5.gif" class="style1">
  回 复:<?php echo $myrow['hfzt'];?></td>
  </tr>
  <tr>
    <td height="100" align="left" valign="top">
<table width="776" height="100" border="0" cellpadding="0" cellspacing="0" background="image/7.gif">
   <?php $query="select * from mr_user where username='".$myrow['username']."'";//获取回复者的相关信息
                        $result=mysql_query($query);
                         $xq=mysql_fetch_array($result);
      ?>
      <tr>
    <td height="24" align="center" valign="middle" class="style1"><?php echo $myrow['username'];?></td>
      </tr>
          <tr>
            <td height="64" align="center" valign="middle">
<img src="<?php echo $xq['tx'];?>"width="60" height="60"></td>
          </tr>
          <tr>
            <td width="80%" height="24" class="style1"> Email:<?php echo $xq['email'];?> 
QQ:<?php echo $xq['qq'];?> 发表时间：<?php echo $myrow['hfsj'];?></td>
```

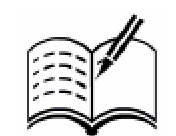




Note

```
                        <td width="20%"> </td>
                      </tr>
                       <tr>
                        <td colspan="2" class="style2">  <?php echo $myrow ['hfnr'];?> </td>
                      </tr>
                    </table></td>
                  </tr>
                </table></td>
            </tr>
                <?php
            }
            ?>
        </table>
      <table width="776" height="40" border="0" align="center" cellpadding="0" cellspacing="0" background="image/4.gif">
            /*........省略部分表单元素......*/
                <td width="45%" class="#ff0000">  页次：<font class="#ff0000"><?php echo $page;?>
        </font> / <font class="#ff0000"><?php echo $page_count;?>
        </font>页 记录：<font class="#ff0000"><?php echo $message_count;?> </font>条 </td>
                  <td width="45%" class="#ff0000">
        分页：<a href='zthf.php?zhuti=<?php echo $zhuti; ?>&recid=<?php echo $_GET['recid'];?>&page=1'>首页</a>
        <?php
                  if($page >= 2){              //如果页数超过 2 页，显示“上一页”超链接
        ?>
                          <a href="zthf.php?zhuti=<?php echo $zhuti; ?>
        &recid=<?php echo $_GET['recid'];?>&page=<?php echo $page-1;?>">上一页</a>
        <?php
                               }
                  if($page < $page_count){    //如果页数在所分页数中，显示“下一页”超链接
        ?>
                          <a href="zthf.php?zhuti=<?php echo $zhuti; ?>
        &recid=<?php echo $_GET['recid'];?>&page=<?php echo $page+1;?>">下一页</a>
        <?php
                  }
                  if($page >= $page_count){//如果页数大于等于所分页数，显示“尾页”超链接
        ?>
                          <a href="zthf.php?zhuti=<?php echo $zhuti; ?>
        &recid=<?php echo $_GET['recid'];?>&page=<?php echo $page_count;?>">尾页</a>
                        <?php
                  }
                          ?>
        /*........省略部分表单元素......*/
        </table>
        <?php
        }else{
              ?>
```





```
<table width="776" height="40" border="0" align="center" cellpadding="0" cellspacing="0" background
="image/4.gif">
      <tr>
        <td height="20" colspan="3">无回复</td>
      </tr>
    </table>
    <?php
    }
    ?>
```

说明：

在 index.php 和 lmzy.php 文件中，使用的方式基本是一致的，这里不再对它们进行介绍，具体的代码可以参见配套资源。

10.4.3　帖子的发布

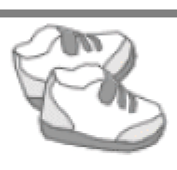
作为本论坛的注册用户，不但可以浏览论坛中的所有帖子，还可以在论坛中相应的专区中发布自己的帖子和对其他的帖子进行回复。这两个模块看似不同，其实使用的技术是同一个道理，都是通过表单的形式把要表达的内容提交到数据处理页中，然后通过数据处理页把信息存储到数据库中。这里以发布帖子的模块为例进行介绍。先来看一下发布帖子页面的运行效果，如图 10.18 所示。



图 10.18　发布帖子

在该页面中，可以设置要发布帖子的类别、主题、表情图标和内容。此外，还有一些看不到的隐藏域，其内容包括发帖人、发帖时间等。其中的表情图标是从数据库中获取的，并且对它们进行了整齐的排列。表情图标排列设计的代码如下：

```
<!--********************************fbzt.php********************************-->
<?php   for($i=1; $i<=4; $i++) {                //如果变量<=4，按照行进行循环，输出行
            echo "<tr>";       ?>
        <?php   if($i==1){
                    $query="select * from mr_lttb ";    //从数据表 mr_lttb 中获取表情图标
                        $lttb=mysql_query($query);
```



Note

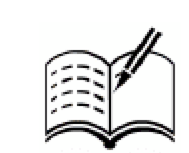
```
                    }
                    $j=1;                                    //设置变量$j=1
                    while ($lttb_row=mysql_fetch_array($lttb)){ //进行 while 循环输出
                        if ($j<=8)   { //每行的表情图标个数<=8，输出如下内容
        ?>
    <td height="26" align="center" valign="middle">
  <input name="xq" type="radio" value="<?php echo $lttb_row['id'];?>" checked="checked">
  <img  src="images.php?recid=<?php  echo  $lttb_row['id'];?>"  width="16"  height="16" border="0"></a></td>
        <?php  }
                $c=1;
                ++$j;                    //当表情图标的个数增加时
                if($j==9){  break;  }    //当表情图标的个数等于 9 时，跳转到下一行重新进行循环
                            }
            echo "</tr>";
        }
  ?>
```

正确填写表单信息后，单击“主题提交”按钮，把信息提交到数据处理页面 fbzt_ok.php 中，最后存储到数据库中。在 fbzt_ok.php 页，通过$_POST 获取表单提交的数据，将数据添加到指定的数据表中。其关键代码如下：

```
<?php
session_start();                                          //初始化 SESSION 变量
include("config.php");                                    //包含数据库连接文件
$time=date("Y")."-".date("m")."-".date("d");              //获取系统当前时间
if(isset($_SESSION['username']) && isset($_SESSION['password'])){//判断用户是否正常登录
    if(isset($_POST['zhuti'])){                           //判断主题是否提交
        $zhuti=htmlspecialchars($_POST['zhuti']);         //对字符串进行格式化
        $neirong=htmlspecialchars($_POST['neirong']);
        $zq=$_POST['zq'];
        $xq=$_POST['xq'];                                 //获取系统提交数据
        $username=$_SESSION['username'];
       $query="insert into mr_zqlb (zq,xq,zhuti,neirong,username,fbsj)
            values('$zq','$xq','$zhuti','$neirong','$username','$time')";
        $result=mysql_query($query,$id);                  //执行添加语句
        if($result){
            echo  "<meta  http-equiv=\"Refresh\"  content=\"2;url=lmzy.php?zq=".urlencode($zq)."\">";
            echo "发布成功!";
        }else{
            echo "发布主题失败!!";
        }
    }
    if(isset($_POST['hftj'])){
        $date=date("y:m:d h:i:s");
       $hfzt=htmlspecialchars($_POST['hfzt']);
        $hfnr=htmlspecialchars($_POST['hfnr']);
        $query="insert into mr_hflb (hfzt,hfnr,hfsj,username,ljid,zq,zhuti)
```



```
            values('$hfzt','$hfnr','$date','".$_SESSION['username']."',
            '".$_POST['ljid']."','".$_POST['zq']."','".$_POST['zt']."')";
        $result=mysql_query($query);                    //送出一个 query 字符串
        if($result){
            echo "回复成功!!";
            echo "<meta http-equiv=\"Refresh\"
                content=\"2;url=zthf.php?zhuti=".$_POST['zt']."&recid=".$_POST ['ljid']."\">";
        }else{
            echo "回复失败!!!";
        }
    }
}else{
    echo "请先登录!!";
}
?>
```



上述代码中不但对发布帖子提交的数据进行了处理，而且还包含了对回复帖子数据进行处理的操作。

技巧：

（1）在设计表单提交页面时，一定要注意在文本域和按钮名称中使用的字符。它是表单提交的关键，如果填写的字符和在数据处理页 fbzt_ok.php 中使用的不一致，后果会很严重，数据将不会被存储到数据库中。

（2）表单中的元素名称与表单处理页使用的名称大小写必须统一，否则在处理页将获取不到提交的值。

（3）在数据处理页 fbzt_ok.php 中，在 insert 语句后输出的字段名称一定要和数据库中的字段名称一致，否则也无法把数据存储到数据库中。

10.4.4　注销用户

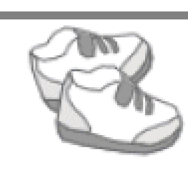
作为注册用户还拥有最后一项权限，那就是注销用户。该权限的操作与其代码一样简单，只需单击“注销”按钮即可退出登录。zx_ok.php 文件的代码如下：

```
<?php
session_start();
session_destroy();
header("location:index.php");
?>
```

10.5　后 台 管 理

10.5.1　在线论坛后台管理概述

在线论坛后台管理的登录入口是论坛头文件中的“版主登录”超链接，通过它进入后





台管理系统（用户名是 mr，密码是 mrsoft）。在后台管理系统中提供了四大功能，分别介绍如下。

- ☑ 栏目管理：用于添加论坛中的专区、版主和对应的图标，同时可以删除指定的专区。
- ☑ 主题管理：分页浏览论坛中发布的主题和删除指定的主题。
- ☑ 回复主题管理：分页浏览论坛中回复帖子的内容和删除指定的回复帖子。
- ☑ 用户管理：分页浏览论坛中注册的用户和删除指定用户。

后台管理系统主页的运行效果如图 10.19 所示。



图 10.19　后台首页

10.5.2　通过 switch 和 include 语句设计后台首页框架

在后台管理系统中，应用 switch 和 include 语句设计一个网页框架，根据超链接传递的参数值，实现在栏目管理、主题管理、回复主题管理和用户管理功能之间的跳转操作。其原理是：首先，分别为不同的功能模块创建超链接，都链接到 index.php 文件，通过超链接传递的参数值来区分不同的功能模块；然后，应用 switch 语句，根据超链接中传递的变量值进行判断，并根据不同的变量值应用 include 语句调用不同的子文件。在线论坛后台首页 index.php 文件的关键代码如下：

```
<?php
session_start();                                        //初始化 SESSION 变量
include("config.php");                                  //包含数据库连接文件
$furl=getenv("http_referer");                           //获取当前文件
$page=(isset($_GET['page']))?$_GET['page']:"1";         //定义分页变量的值
$lmbs=(isset($_GET['lmbs']))?$_GET['lmbs']:"";          //定义栏目标识传递的参数值
if(isset($_SESSION['user']) && isset($_SESSION['pass'])){ //判断当前用户的权限
?>
<table width="175" border="0" cellpadding="0" cellspacing="0">
    <tr>
     <td height="30" ><a href="index.php?lmbs=<?php echo urlencode('栏目管理');?>">栏目管理</a></td>
      </tr>
      <tr>
```



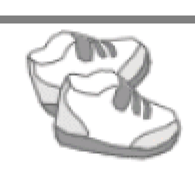


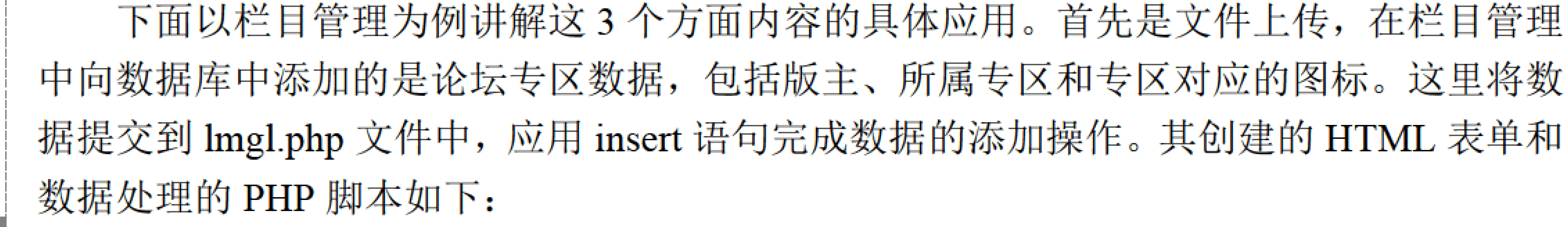

```
        <td height="30" ><a href="index.php?lmbs=<?php echo urlencode('论坛主题');?>">主题管理</a></td>
            </tr>
            <tr>
        <td height="30" ><a href="index.php?lmbs=<?php echo urlencode('回复主题');?>">回复主题管理</a></td>
            </tr>
            <tr>
        <td height="30" ><a href="index.php?lmbs=<?php echo urlencode('用户管理');?>">用户管理</a></td>
            </tr>
        </table>
<?php
switch($lmbs){                          //通过该语句来调用相应的文件
        case "栏目管理":
            include "lmgl.php";         //如果单击的超链接等于栏目管理，则输出 lmgl.php 页
        break;                          //如果不是，则跳出，继续其他的循环
        case "论坛主题":
            include "ztgl.php";
        break;
        case "回复主题":
            include "hfztgl.php";
        break;
        case "用户管理":
            include "hygl.php";
        break;
        default:
            include "lmgl.php";
        break;
}
?>
<!--省略部分代码-->
<?php
}else{                                  //判断当前用户的权限
    echo "<meta http-equiv=\"Refresh\" content=\"2;url=admin.php\">";
    echo "<a href=\"admin.php\">这里</a>";
}
?>
```

10.5.3 栏目管理

后台管理系统中提供了栏目管理、主题管理、回复主题管理以及用户管理四大功能。对于这些功能，可以将其总结为 3 个方面的内容：一是上传文件；二是从数据库中查询数据，完成数据的分页输出；三是删除指定的数据。

所谓上传就是通过表单提交，使用 insert 语句把数据添加到数据库中；而查询和分页输出数据，则是使用 select 语句从数据库中根据不同的条件进行查询，从而输出相应的内容；删除操作是在使用 select 语句查询出数据的基础上，使用 delete 语句以指定的 ID 值为







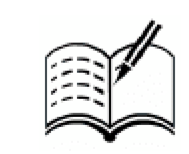
Note

条件完成的。

下面以栏目管理为例讲解这 3 个方面内容的具体应用。首先是文件上传，在栏目管理中向数据库中添加的是论坛专区数据，包括版主、所属专区和专区对应的图标。这里将数据提交到 lmgl.php 文件中，应用 insert 语句完成数据的添加操作。其创建的 HTML 表单和数据处理的 PHP 脚本如下：

```
<table width="590" border="1" cellpadding="0" cellspacing="0">
<form name="myform" method="post" action="index.php?lmbs=栏目管理" enctype="multipart/form-data">
    <tr>
            <td height="30" colspan="2" align="center">栏目管理</td>
            <td width="220"> </td>
    </tr>
    <tr>
            <td width="170" height="30" align="center"><span class="style1">版主:
            <input name="bz" type="text" id="bz" size="15">
            </span></td>
            <td width="200" align="center" class="style1">所属专区:
         <select name="zq" size="1" id="zq">
           <option value="asp" selected>ASP</option>
           <option value="jsp">JSP</option>
           <option value="delphi">Delphi</option>
           <option value="visual basic">Visual Basic</option>
           <option value="visual foxpro">Visual Foxpro</option>
           <option value="visual c++">Visual C++ </option>
           <option value="power">Power Buider</option>
           <option value=".net">.net</option>
        </select></td>
            <td align="center" valign="middle" class="style1">图标:
            <input type="radio" name="tb" value="<?php echo $top;?>">
            <img src='../image/tx/top.gif' width='24' height='24'>
            <input type="radio" name="tb" value="<?php echo $df;?>">
            <img src='../image/tx/df.gif' width='24' height='24'>
            </td>
    </tr>
    <tr>
            <td colspan="2"> </td>
            <td class="style1"><input name="zhuijia" type="submit" id="zhuijia" value="追加栏目"></td>
    </tr>
</form>
</table>
<?php
if(isset($_SESSION['user']) && isset($_SESSION['pass'])){
    $top='image/tx/top.gif';                          //通过变量来获取专区的图标
    $df='image/tx/df.gif';
    $date=date("Y-m-d h:i:s");                        //获取当前的时间
    if(isset($_POST['zhuijia'])){                     //把提交的数据添加到数据库中
        $query="insert into mr_zqfl(bz,zq,tb,cjsj)
            values('".$_POST['bz']."','".$_POST['zq']."','".$_POST['tb']."','".$date."')";
```






```
            $result=mysql_query($query);
            if($result){
                echo "<meta http-equiv=\"refresh\" content=\"0;url=$furl\">";
            }else{
                echo "追加失败!!";
            }
        }
?>
```

然后，在 lmgl.php 文件中，连接数据库，通过 select 语句查询数据库中存储的论坛专区数据，应用 while 语句循环输出查询结果，并且创建“删除”超链接，链接到 delete.php 文件，以论坛专区的 ID 值为参数。其关键代码如下：

```
<table width="590" border="0" cellpadding="0" cellspacing="0">
    <tr align="center" bgcolor="#D0E8FF" class="style1">
    <td width="95" height="30">图  标</td>
    <td width="70">所属专区</td>
    <td width="145" bgcolor="#D0E8FF">版  主</td>
    <td width="200">时间</td>
    <td width="80">是否删除</td>
    </tr>
<?php
$query="select * from mr_zqfl where id";      //从数据库中查询出有关栏目专区的信息
$result=mysql_query($query);
if($result){
    while($myrow=mysql_fetch_array($result)){ //通过 while 循环语句把这些信息的内容进行输出
?>
    <tr align="center" class="style1">
    <td height="30"><img src=../<?php echo $myrow['tb'];?> /></td>
    <td><?php echo $myrow['zq'];?></td>
    <td><?php echo $myrow['bz'];?></td>
    <td><?php echo $myrow['cjsj'];?></td>
    <td>
    <a href="delete.php?lmbs=<?php echo urlencode('栏目管理');?>&id=<?php echo $myrow['id'];?>">删除</a>
        </td>
    </tr>
<?php
    }
}
?>
</table>
```

最后，创建 delete.php 文件，以超链接传递的值为条件，执行 delete 语句删除指定的论坛专区数据。其关键代码如下：

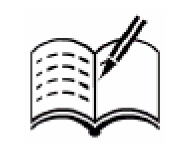
```
<?php
session_start();
include("config.php");
$furl=getenv("http_referer");
```

```
if(isset($_SESSION['user']) && isset($_SESSION['pass'])){
    if(isset($_GET['lmbs']) && $_GET['lmbs']=="用户管理"){
        $id=$_GET['id'];                                  //获得将被删除的用户 ID
        $query="delete from mr_user where id='$id'"; //将要删除的用户从 mr_user 表中删除
        $result=mysql_query($query);
        if($result){
            echo "删除成功!!";
            echo "<meta http-equiv=\"Refresh\" content=\"2;url=$furl\">";
        }else{
            echo "删除失败!";
        }
    }
    if(isset($_GET['lmbs']) && $_GET['lmbs']=="栏目管理"){
        $id=$_GET['id'];
        $query="delete from mr_zqfl where id='$id'";
        $result=mysql_query($query);
        if($result){
            echo "<font color=\"#ff0000\">删除成功!!</font>";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=$furl\">";
        }
    }
    if(isset($_GET['lmbs']) && $_GET['lmbs']=="论坛主题"){
        $id=$_GET['id'];
        $query="delete from mr_zqlb where id='$id'";
        $result=mysql_query($query);
        if($result){
            echo "成功删除!!";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=$furl\">";
        }
    }
    if(isset($_GET['lmbs']) && $_GET['lmbs']=="回复主题"){
        $id=$_GET['id'];
        $query="delete from mr_hflb where id='$id'";
        $result=mysql_query($query);
        if($result){
            echo "<font color=\"#ff0000\">删除成功!!</font>";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=$furl\">";
        }
    }
}else{
    echo "<meta http-equiv=\"Refresh\" content=\"2;url=admin.php\">";
    echo "<a href=\"admin.php\">这里</a>";
}
?>
```

说明：

在 delete.php 文件中，存储了后台管理系统中所有删除操作的代码。其根据超链接传递的参数值进行判断，具体执行哪个删除语句。





10.6　技 术 提 炼

在线论坛系统中用到了分页、网页访问权限的设置和页面刷新等技术。下面对这些技术的应用进行提炼、总结。

10.6.1　分页技术

众所周知，网站的分页方法很多。下面介绍笔者在本论坛中使用的分页技术是如何实现的。

在线论坛中的专区帖子页（lmzy.php 文件）中使用了分页技术，实现了输出页次、页数、共有多少条记录。此外，在每一页中显示一条记录，然后通过“首页”、“上一页”、“下一页”以及“尾页”超链接来实现分页的输出。分页的运行效果如图 10.20 所示。



图 10.20　专区帖子

分页技术的设计思路：首先使用 select 查询语句，从数据库中查询出发布的帖子的所有记录；然后根据分页超链接传递的分页变量值，以及查询出的总的记录数，计算出当前页输出数据的开始位置；接着，重新定义 select 查询语句，应用 limit 关键字控制查询数据的开始位置和查询的记录数；最后，通过 while 语句循环输出查询结果。其具体操作代码如下（lmzy.php 文件）：

```
<?php
include("config.php");
$page=(isset($_GET['page']))?$_GET['page']:"1";   //获取分页超链接传递的分页变量值
$zq=(isset($_GET['zq']))?$_GET['zq']:"PHP";
?>
<?php
$page_size=1;                                       //每页显示 1 条记录
$query="select * from mr_zqlb where zq='".$zq."'"; //从数据库中查询所有符合条件的数据
$result=mysql_query($query);
$message_count=mysql_num_rows($result);            //统计总的记录数
if($message_count>0){
?>
<table width="776" height="30"  border="1" align="center" cellpadding="1" cellspacing="1">
  <tr align="center">
    <td width="51" height="26" class="style1">状态</td>
```



Note

```
        <td width="54" class="style1">心情</td>
        <td width="381" class="style1">主      题</td>
        <td width="79" class="style1">作者</td>
        <td width="76" class="style1">回复/人气</td>
        <td width="135" class="style1">发表时间 </td>
    </tr>
<?php
$page_count=ceil($message_count/$page_size);                //计算总的页码数
$offset=($page-1)*$page_size;                               //计算当前页查询的开始位置
$query="select * from mr_zqlb where zq='".$zq."' order by id desc limit $offset ,$page_size ";
$result=mysql_query($query);                                //查询当前页显示的记录
while($myrow=mysql_fetch_array($result)){                   //循环输出查询结果
?>
 <!--省略部分代码-->
<?php
}
?>
</table>
```

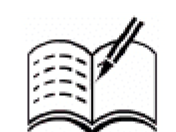
上述代码完成了分页查询操作，以及查询结果的循环输出。下面介绍分页超链接的创建，其关键代码如下：

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr class="style1">
<td width="40%">页次：<?php echo $page;?>/<?php echo $page_count;?>页 记录：<?php echo $message_count;?>条</td>
        <td width="50%"> 分页：
            <a href='lmzy.php?zq=<?php echo urlencode($zq);?>&page=1'>首页</a>
<?php
        if($page >= 2){
?>
        <a href="lmzy.php?zq=<?php echo urlencode($zq);?>&page=<?php echo $page-1;?>">上一页</a>
<?php
        }
        if($page < $page_count){
?>
        <a href="lmzy.php?zq=<?php echo urlencode($zq);?>&page=<?php echo $page+1;?>">下一页</a>
<?php
        }
        if($page <= $page_count){
?>
        <a href="lmzy.php?zq=<?php echo urlencode($zq);?>&page=<?php echo $page_count;?>">尾页</a>
<?php
        }
?>
        </td>
    </tr>
```



```
</table>
```

10.6.2　网页访问权限设置



网页访问权限设置的目的是控制不同用户拥有不同的权限。例如，在后台管理系统中，只有管理员才具备访问权限，可以执行各种操作，其他用户不能访问后台管理系统的网页；而在论坛系统中，只有登录成功的会员才具备发布和回复帖子的权限，普通游客不具备此权限。那么这个网页访问的权限是如何设置的呢？

这里以后台管理系统中的网页为例，讲解如何设置网页访问权限。其关键是会话（SESSION）的应用，它解决了 HTTP Web 协议的无状态问题，实现数据在不同页面之间的传递。在此设置网页的访问权限就是通过判断 SESSION 变量的值来完成的。

首先，在后台管理员登录成功后，将管理员的用户名和密码存储到 SESSION 变量中。然后，在后台管理系统的网页中，通过$_SESSION 获取 SESSION 变量中存储的用户名和密码，通过 isset()函数判断其是否存在，如果存在则说明当前用户可以访问此网页；否则当前用户就不可以访问此网页，给出提示信息，跳转到后台管理系统的登录页。其关键代码如下：

```
<?php
session_start();                                                    //初始化 SESSION 变量
if(isset($_SESSION['user']) && isset($_SESSION['pass'])){           //判断当前用户的权限
?>
<!--省略了部门代码-->
<?php
}else{
        echo "<meta http-equiv=\"Refresh\" content=\"2;url=admin.php\">";
        echo "<a href=\"admin.php\">这里</a>";
}
?>
```

10.6.3　页面刷新技术

为了能够更好地浏览到论坛中的信息，笔者在论坛中采用了页面刷新技术，以此来保持页面中始终输出最新的数据。此技术的实现应用的是一个简单的 JavaScript 语句，其代码如下：

```
<a href="javascript:location.reload()" target="_self">刷新页面</a>
```

通过单击“刷新页面”超链接，即可完成对页面数据的更新操作。

此外，还有一种刷新页面的方法，可以控制刷新的时间，即每隔一段时间对页面刷新一次。其代码如下：

```
<meta http-equiv="refresh" content="2;url=index.php">
```

上述代码的含义是每隔 2 秒钟对 index.php 页刷新一次。



博客模块

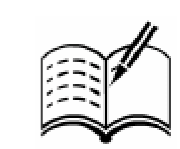
（自学视频、源程序：配套资源\mr\11\）

如果有人问："什么是Web 2.0？"，恐怕没几个人能说得清楚，但是要问"什么是博客？"，哪怕是从不上网的人也耳熟能详。博客正是Web 2.0概念中重要的组成部分之一（其他众所周知的还包括IM即时通和RSS阅读器）。

Blog（博客），全名Weblog，后来缩写为Blog。Blogger就是写Blog的人，习惯于在网上写日记、发布个人照片、展示个性自我的用户群体。对于Blog/Blogger的中文名称，有翻译成"博客"的，也有翻译为"网志"的，但大多数人都已经认可了"博客"。通过阅读本章，读者可以学到以下内容。

- 系统设计思路
- 数据库设计
- 博客空间个人首页设计
- 我的文章模块的设计
- 文章浏览模块设计
- 文章管理模块设计
- 好友管理模块设计
- 小纸条管理模块设计
- 用户管理模块设计





Note

11.1 博 客 概 述

11.1.1 模块概述

博客，即在网络上发表文章、展现个性的一类人；而博客系统，是为这类人群提供一个相互交流、学习的平台。博客的两大基本功能是共享与交流。共享，就是将文章、图片、心得等一些在传统概念中属于私人的东西，拿出来和多数人一起分享；交流，就是有着同样兴趣、爱好、语言的一类人之间的联系。

11.1.2 功能结构

本章的博客系统主要分为个人博客管理和博客后台管理两部分。

☑ 个人博客管理：主要功能包括个人管理、文章管理、相册管理、好友管理、留言管理和小纸条管理。个人博客管理功能结构如图 11.1 所示。

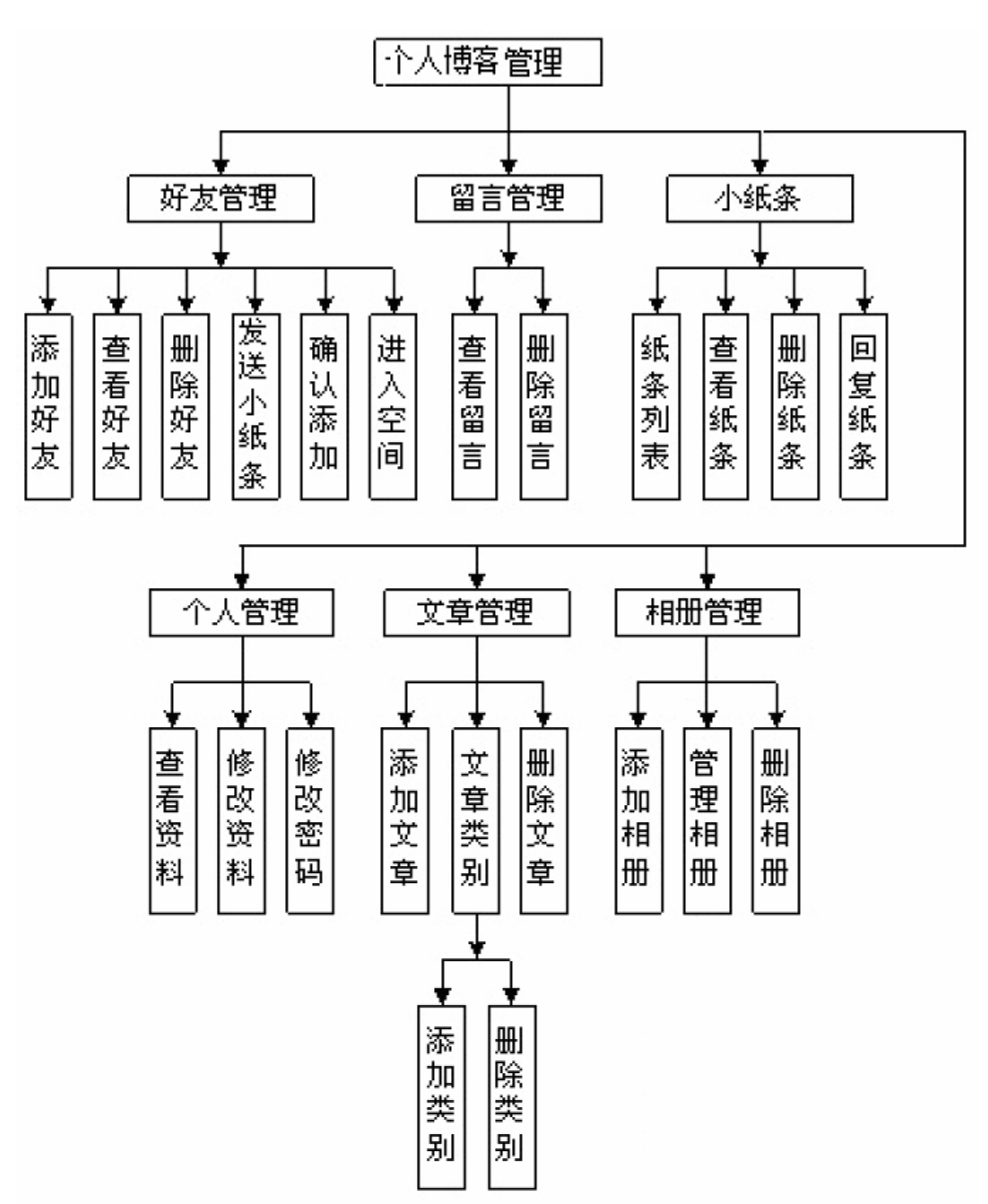
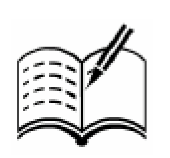


图 11.1 个人博客管理功能结构图

☑ 博客后台管理：主要是对博客用户和管理员的管理。其主要功能包括用户管理、用户文章管理、用户相册管理、管理员管理、系统日志管理和数据备份、恢复。





Note

博客后台管理的功能结构如图 11.2 所示。

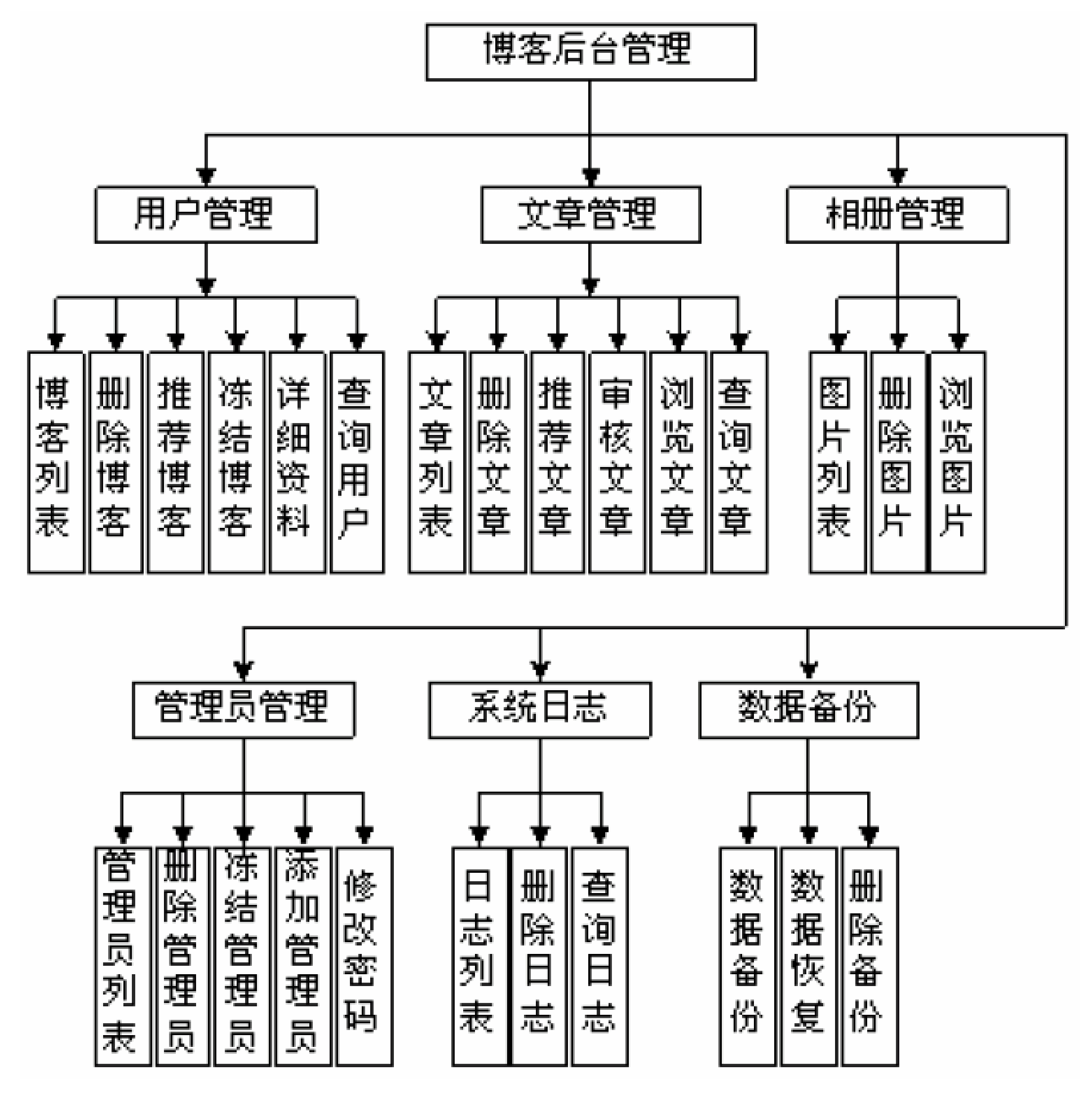


图 11.2　博客后台管理功能结构图

11.1.3　程序预览

为了让读者对本模块有个初步的了解和认识，下面给出其中几个页面运行效果图；如果想查看完整的效果图，参见配套资源源程序。

博客首页运行效果如图 11.3 所示。



图 11.3　博客首页运行效果



个人博客管理页面运行效果如图 11.4 所示。



图 11.4　个人博客管理页面

博客用户管理页面运行效果如图 11.5 所示。



图 11.5　博客用户管理页面运行效果

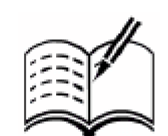
图片管理页面运行效果如图 11.6 所示。



图 11.6　图片管理页面运行效果

Note





11.2 数据库设计

本模块属于中小型个人网站，毫无争议地，依然采用 PHP+MySQL 这对黄金组合（无论是从成本、性能、安全，还是从易操作性上考虑，MySQL 都是最佳选择）。

11.2.1 创建数据库与数据表

博客系统使用的数据库是 db_blog，其中包含 9 张数据表，各表的名称和作用如图 11.7 所示。



服务器: localhost ▸ 数据库: db_blog

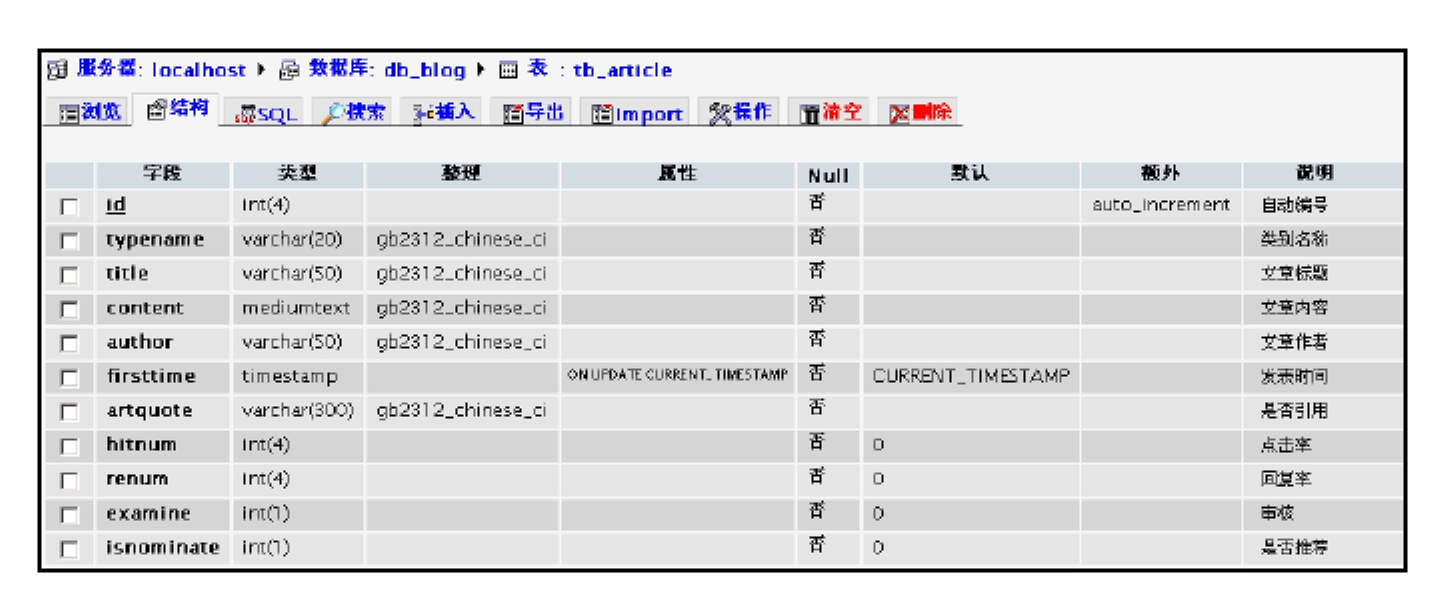
表	操作	记录数	类型	整理	说明
tb_admin		1	MyISAM	gb2312_chinese_ci	管理员信息表
tb_article		10	MyISAM	gb2312_chinese_ci	博客文章列表
tb_frd		1	MyISAM	gb2312_chinese_ci	用户好友信息表
tb_log		127	MyISAM	gb2312_chinese_ci	文章日志信息表
tb_member		12	MyISAM	gb2312_chinese_ci	用户信息数据表
tb_mess		17	MyISAM	gb2312_chinese_ci	用户留言数据表
tb_review		31	MyISAM	gb2312_chinese_ci	文章评论表
tb_script		13	MyISAM	gb2312_chinese_ci	小纸条列表
tb_uppics		4	MyISAM	gb2312_chinese_ci	上传图片信息
9 个表	总计	216	MyISAM	gb2312_chinese_ci	

图 11.7 数据表列表

下面来具体看一下其中几张重要数据表的结构设计。

1. tb_article（博客文章列表）

tb_article 表中包含文章的类别、文章标题、文章内容等信息，其结构及字段说明如图 11.8 所示。



服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_article

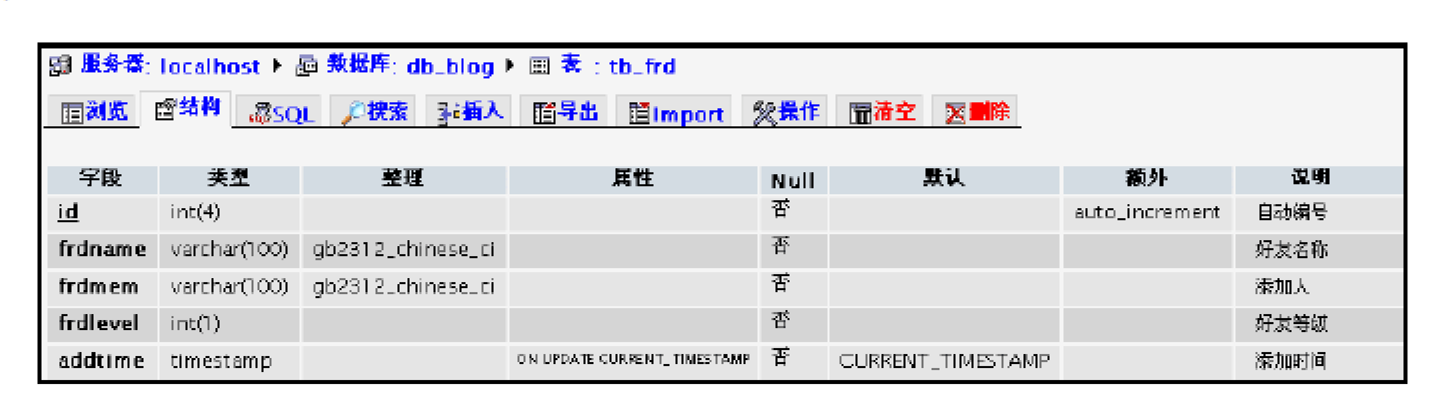
浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
typename	varchar(20)	gb2312_chinese_ci		否			类别名称
title	varchar(50)	gb2312_chinese_ci		否			文章标题
content	mediumtext	gb2312_chinese_ci		否			文章内容
author	varchar(50)	gb2312_chinese_ci		否			文章作者
firsttime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		发表时间
artquote	varchar(300)	gb2312_chinese_ci		否			是否引用
hitnum	int(4)			否	0		点击率
renum	int(4)			否	0		回复率
examine	int(1)			否	0		审核
isnominate	int(1)			否	0		是否推荐

图 11.8 tb_article 表结构

2. tb_frd（用户好友信息表）

tb_frd 表中包含好友名称、添加人、好友等级和添加时间等字段，其结构及字段说明如图 11.9 所示。



服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_frd

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

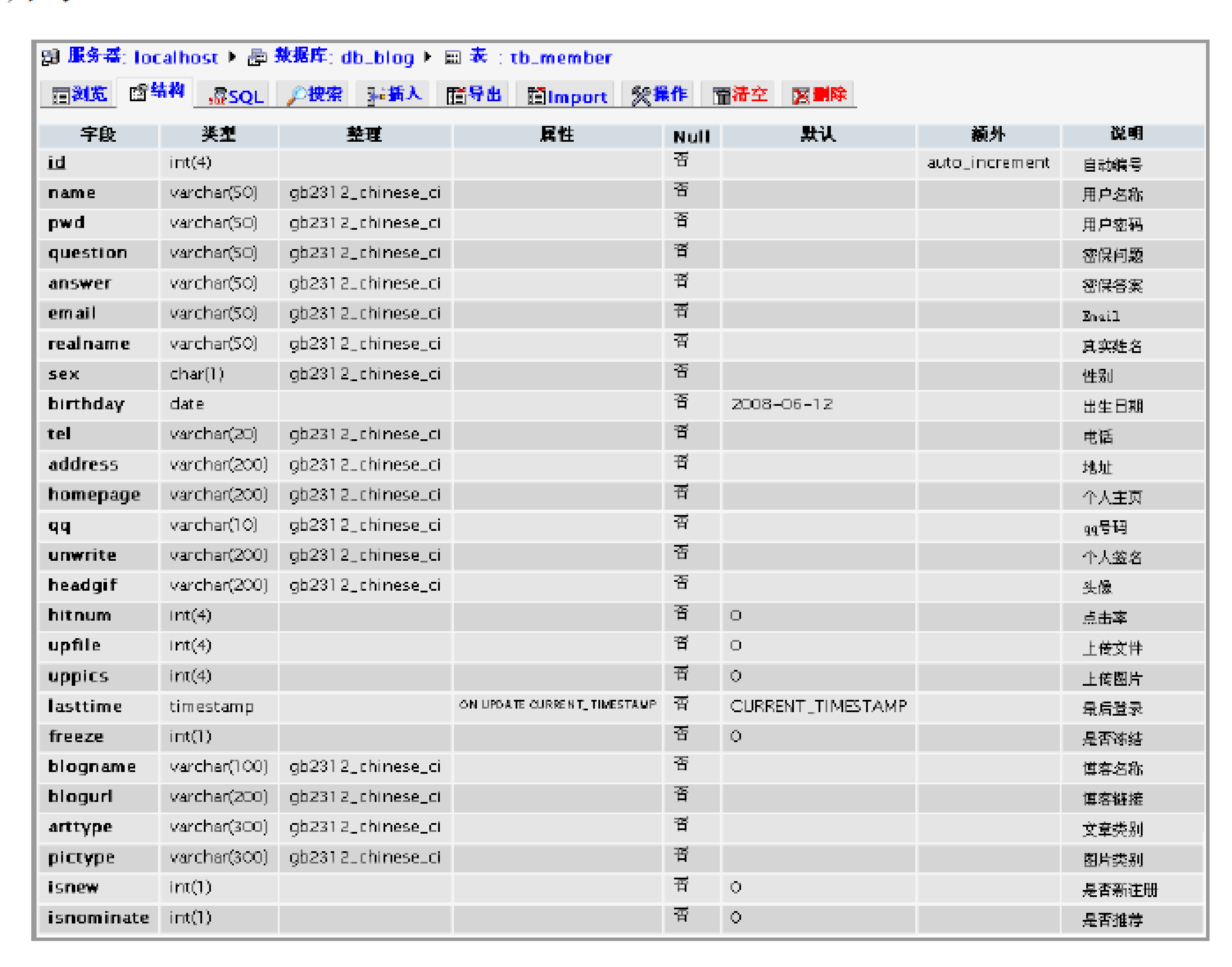
字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
frdname	varchar(100)	gb2312_chinese_ci		否			好友名称
frdmem	varchar(100)	gb2312_chinese_ci		否			添加人
frdlevel	int(1)			否			好友等级
addtime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		添加时间

图 11.9 tb_frd 表结构



3. tb_member（用户信息数据表）

tb_member 表中包含用户的基本资料，如 E-mail、姓名、出生日期等，以及个人博客中的一些参数，如点击率、文章类别、相册类别、是否推荐等。tb_member 表结构及说明如图 11.10 所示。



服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_member

浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	gb2312_chinese_ci		否			用户名称
pwd	varchar(50)	gb2312_chinese_ci		否			用户密码
question	varchar(50)	gb2312_chinese_ci		否			密保问题
answer	varchar(50)	gb2312_chinese_ci		否			密保答案
email	varchar(50)	gb2312_chinese_ci		否			Email
realname	varchar(50)	gb2312_chinese_ci		否			真实姓名
sex	char(1)	gb2312_chinese_ci		否			性别
birthday	date			否	2008-06-12		出生日期
tel	varchar(20)	gb2312_chinese_ci		否			电话
address	varchar(200)	gb2312_chinese_ci		否			地址
homepage	varchar(200)	gb2312_chinese_ci		否			个人主页
qq	varchar(10)	gb2312_chinese_ci		否			qq号码
unwrite	varchar(200)	gb2312_chinese_ci		否			个人签名
headgif	varchar(200)	gb2312_chinese_ci		否			头像
hitnum	int(4)			否	0		点击率
upfile	int(4)			否	0		上传文件
uppics	int(4)			否	0		上传图片
lasttime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		最后登录
freeze	int(1)			否	0		是否冻结
blogname	varchar(100)	gb2312_chinese_ci		否			博客名称
blogurl	varchar(200)	gb2312_chinese_ci		否			博客链接
arttype	varchar(300)	gb2312_chinese_ci		否			文章类别
pictype	varchar(300)	gb2312_chinese_ci		否			图片类别
isnew	int(1)			否	0		是否新注册
isnominate	int(1)			否	0		是否推荐

图 11.10　tb_member 表结构

4. tb_review 表（评论列表）

tb_review 表中包含被评论的文章 id、发表评论人、评论内容以及评论时间，其结构如图 11.11 所示。



服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_review

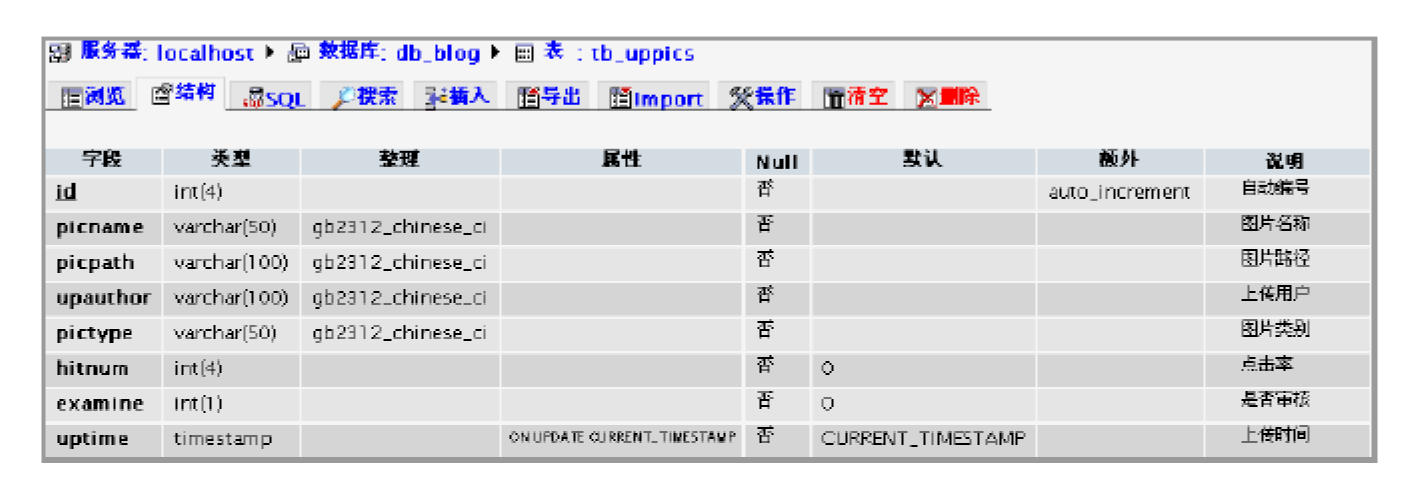
浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
artid	int(4)			否			文章id
content	varchar(300)	gb2312_chinese_ci		否			评论内容
man	varchar(100)	gb2312_chinese_ci		否			评论人
firsttime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		评论时间

图 11.11　tb_review 表结构

5. tb_uppics 表（上传图片列表）

tb_uppics 表中包含上传图片的信息，如上传图片名称、上传用户、上传类别等，其结构如图 11.12 所示。

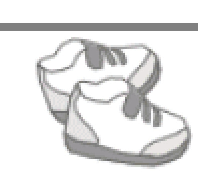


服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_uppics

浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
picname	varchar(50)	gb2312_chinese_ci		否			图片名称
picpath	varchar(100)	gb2312_chinese_ci		否			图片路径
upauthor	varchar(100)	gb2312_chinese_ci		否			上传用户
pictype	varchar(50)	gb2312_chinese_ci		否			图片类别
hitnum	int(4)			否	0		点击率
examine	int(1)			否	0		是否审核
uptime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		上传时间

图 11.12　tb_uppics 表结构





Note

此外，还有管理员信息表、文章日志信息表、用户留言数据表、小纸条列表，限于篇幅，这里不再介绍，读者可参见配套资源中的数据库文件。

11.2.2 数据库操作类

为了方便使用、减少代码，本模块采用 MySQLI 扩展存取数据。与 MySQL 扩展相比，MySQLI 扩展在以下方面有了明显的提高。

- ☑ 兼容性与维护性：MySQLI 拥有比 MySQL 更高的兼容性，MySQLI 扩展可以很容易地使用 MySQL 的新功能，即使 MySQL 的新版本又出现了更多功能，MySQLI 扩展也可以很容易地支持。
- ☑ 面向对象：MySQLI 扩展已封装到一个类中，从而可使用面向对象的方式编程。即使对面向对象不了解，MySQLI 也可以利用面向过程的编程方式来实现。
- ☑ 速度和安全性：MySQLI 扩展执行的速度要比之前版本的 MySQL 扩展快了很多。MySQLI 扩展支持 MySQL 新版本的密码杂凑（Password Hashes）和验证程序，大大提高了应用程序的安全性。
- ☑ 预准备语句：预准备语句可提高重复使用的语句的性能，MySQLI 扩展提供了对预准备语句的支持。
- ☑ 调试功能：MySQLI 扩展进一步改进了调试功能，提高了开发效率。要在 PHP 中使用 MySQLI 扩展，需要在配置文件 php.ini 中添加“extension=php_mysqli.dll”设置；如果配置文件中已有上述设置，则应确保 extension 的前面没有“;”，否则将其去掉。

下面介绍如何使用 MySQLI 扩展来存取数据库。MySQLI 提供了面向对象和面向过程两种方式来与数据库交互。在本模块中，后台管理应用面向对象方式与数据库交互，前台应用面向过程方式与数据库交互。

下面是后台管理 MySQLI 扩展应用的面向对象操作类。在这里连接数据库就是实例化 MySQLI 系统内定的一个类，将所得到的返回集“$this->conn”以“$this->result = $this->conn->query($sql)”的形式写入。其存储于 admin\conn\conn.php 文件中。步骤如下：

（1）定义私有属性，如主机名、用户名、用户密码、数据库名等。

```
<?php
class opmysqli{
        private $host = '127.0.0.1';                    //服务器地址
        private $name = 'root';                         //登录账号
        private $pwd = '111';                           //登录密码
        private $dBase = 'db_blog';                     //数据库名称
        private $conn = '';                             //数据库连接资源
        private $result = '';                           //结果集
        private $msg = '';                              //返回结果
        private $fields;                                //返回字段
        private $fieldsNum = 0;                         //返回字段数
        private $rowsNum = 0;                           //返回结果数
        private $filesArray = array();                  //返回字段数组
```



```
		private $rowsArray = array();				//返回结果数组
```

（2）使用__construct 关键字初始化类。这与 MySQL 扩展写法一样。

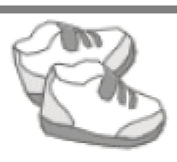
```
		function __construct($host='',$name='',$pwd='',$dBase=''){			//初始化类
			if($host != '')
				$this->host = $host;
			if($name != '')
				$this->name = $name;
			if($pwd != '')
				$this->pwd = $pwd;
			if($dBase != '')
				$this->dBase = $dBase;
			$this->init_conn();
		}
```

（3）连接数据库需要实例化 MySQLI 类。这个类是 PHP 环境自带的，所以不要管它是从哪里来的，只需实例化即可。

```
	function init_conn(){
			$this->conn=new mysqli($this->host,$this->name,$this->pwd,$this->dBase);
			$this->conn->query("set names gb2312");
		}
```

（4）下面是一些具体的操作方法，包括查询结果、取得字段数、更新、删除、添加记录数等。其部分代码如下：

```
	//查询结果
		function mysqli_query_rst($sql){
			if($this->conn == ''){
				$this->init_conn();
			}
			$this->result = @$this->conn->query($sql);
		}
		//取得字段数
		function getFieldsNum($sql){
			$this->mysqli_query_rst($sql);
			$this->fieldsNum = @$this->result->num_fields($this->result);
		}
		//取得查询结果数
		function getRowsNum($sql){
			$this->mysqli_query_rst($sql);
			$this->rowsNum = @$this->result->num_rows;
			return $this->rowsNum;
		}
		//取得结果数值
		function getRowsArray($sql){
			$this->mysqli_query_rst($sql);
			while($row = @$this->result->fetch_array(MYSQLI_ASSOC)) {
			$this->rowsArray[] = $row;
			}
```





```
            return $this->rowsArray;
        }
        //更新、删除、添加记录数
        function uidRst($sql){
            if($this->conn == ''){
                $this->init_conn();
            }
            $this->conn->query($sql);
            $this->rowsNum = @$this->conn->affected_rows;
            return $this->rowsNum;
        }
```

（5）最后实例化 opmysqli 类即可。

```
$conne = new opmysqli();
?>
```

前台使用 MySQLI 以面向过程方式与数据库进行交互。其使用方法类似于 MySQL 扩展。例如，mysql_query 这个 MySQL 扩展函数，在 MySQLI 扩展中写成 mysqli_query。返回结果集在 MySQL 中是 mysql_query($sql，$conn)，而在 MySQLI 扩展中则是 mysqli_query($conn,$sql)。下面给出 MySQLI 面向过程操作数据库的部分代码，其存储于 center/conn/conn.php 文件中。

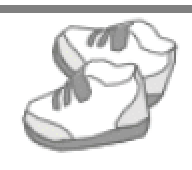
```
/连接数据库
        function init_conn(){
            $this->conn=mysqli_connect($this->host,$this->name,$this->pwd,$this->dBase);
            mysqli_query($this->conn,"set names gb2312");
        }
//查询结果
        function mysqli_query_rst($sql){
            if($this->conn == ''){
                $this->init_conn();
            }
            $this->result = @mysqli_query($this->conn,$sql);
        }
        //取得字段数
        function getFieldsNum($sql){
            $this->mysqli_query_rst($sql);
            $this->fieldsNum = @mysqli_num_fields($this->result);
        }
```

至此，有关数据库操作类的讲解完毕，其具体代码参见配套资源中内容。

11.3 博客空间个人首页

11.3.1 博客空间个人首页概述

博客系统的模块较多，限于篇幅，这里只对重点模块进行讲解。首先来看个人首页。



个人首页由多个页面组成，如热门文章、文章分类、留言列表等。其页面的设计结构如图 11.13 所示。

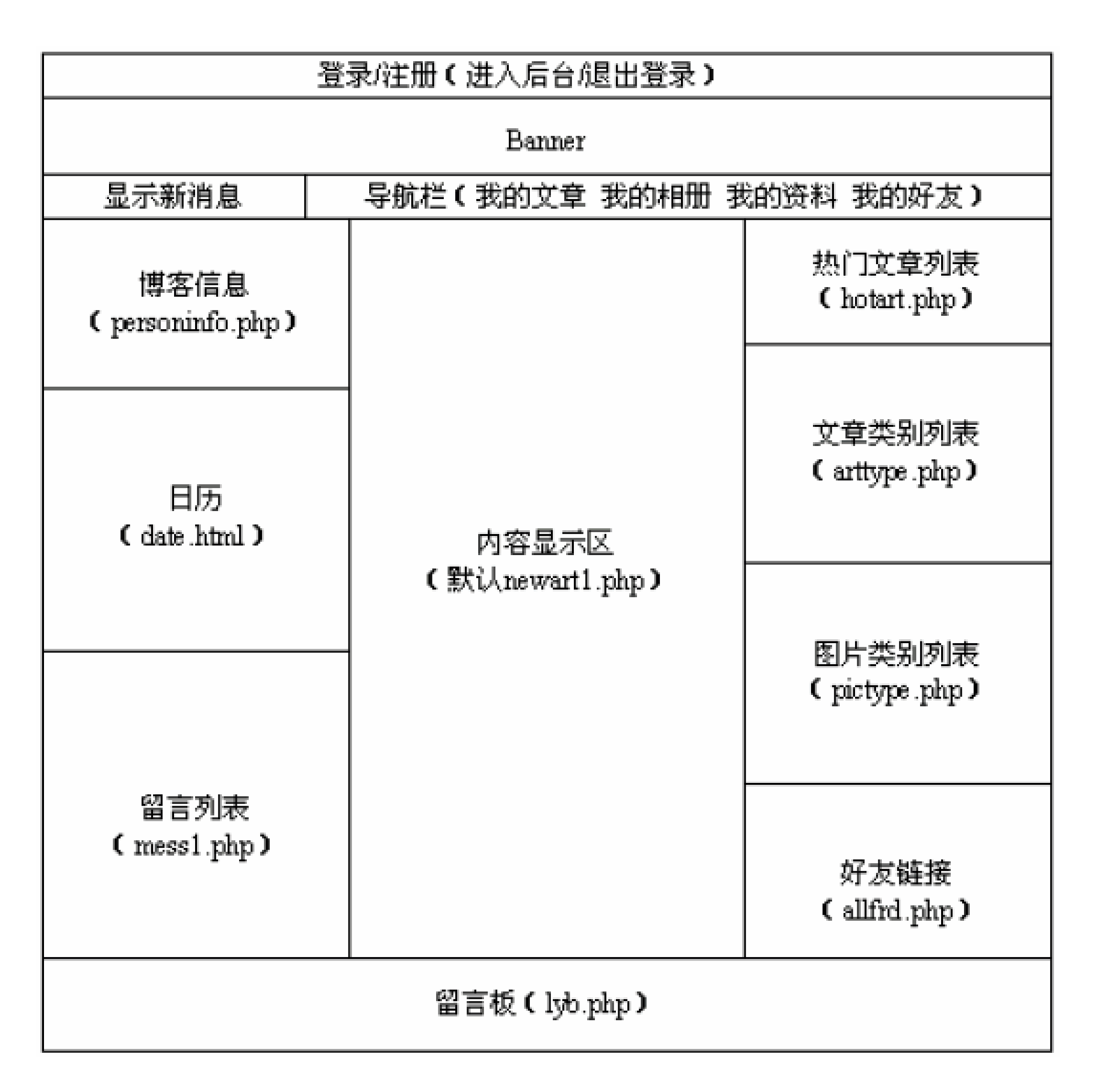


图 11.13　博客空间个人首页结构及布局

个人首页的运行效果如图 11.14 所示。



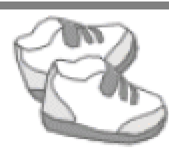
图 11.14　个人首页运行效果

11.3.2　首页实现过程

个人首页设计非常简单，总体上采用 DIV+CSS 来实现（如导航条、个人首页的头部，还有外层等），然后运用包含文件结合技术，把所有需要显示在首页的文件和配置文件引入进来就可以了。其具体步骤如下：

（1）在 center\center.php 文件中，加载各种文件，如配置文件、数据库类文件、计数统计文件、CSS 样式文件、JS 脚本文件等。代码如下：

```
<?php
    session_start();                                    //开启 session
```




```
        header('Content-Type:text/html;charset=gb2312');          //设置页面编码
        include_once '../config.php';                            //加载配置文件
        include_once 'conn/conn.php';                            //加载数据库类
        include_once 'inc/count.php';                            //加载计数器
?>
<link href="../css/style.css" rel="stylesheet" type="text/css" />
<script language="javascript" src="../js/xmlhttprequest.js"></script>
<script language="javascript" src="../js/center.js"></script>
<script language="javascript" src="../js/date.js"></script>
<script language="javascript" src="../js/personinfo.js"></script>
…
```

（2）创建“登录”、“注册”超链接。判断当前的 SESSION 值是否为空，如果为空，就显示“登录”、“注册”超链接；否则，显示光临信息和进入后台管理的超链接。代码如下：

```
  <?php
      if(!empty($_SESSION['name']) and !is_null($_SESSION['name'])){
?>
      您好，欢迎光临<?php echo $_SESSION['name']; ?>的博客！
<a id='blogmain' onclick="javascript:open('manage.php','_parent','',false)">进入管理页面</a> 
<a onclick="javascript:open('logout.php','_parent','',false)">退出登录</a> 
      <?php
       }else{
?>
      <a  onclick="javascript:window.open('login.php','login','width=300,height=200',false)">登录</a>
      <a onclick="javascript:window.open('register.php','register','',false)">注册</a> 
      <?php
}
?>
```

（3）创建 Banner，并且输出博客空间的名称。Banner 下面的一行由两部分构成，一部分是新消息提醒，另一部分是导航栏。新消息提醒每隔一段时间，查看一下个人的纸条信息，如果发现新纸条，会马上显示出来。代码如下：

```
<div id="nav">
      <div id="newscrip"> 
         <script>setInterval("shownewmess()",10000);</script>
      </div>
      <div id="rightfloat">
<a  onclick='javascript:open("center.php?uid=<?php echo $_GET['uid'] ?>","_parent","",false)'>我的文章</a>
<a onclick='javascript:open("center.php?act=pics&uid=<?php echo $_GET['uid'] ?> ","_parent","",false)'>我的相册</a>
<a onclick='javascript:open("center.php?act=person&uid=<?php echo $_GET['uid'] ?> ","_parent","",false)'>我的资料</a>
<a onclick='javascript:open("center.php?act=frd&uid=<?php echo $_GET['uid'] ?>","_parent","",false)'>我的好友</a></div>
   </div>
```



（4）加载各个页面。其中，内容显示区是首页的核心，几乎所有博客空间的功能，都将在内容显示区显示。内容显示区的代码如下：

```
…
        <?php
            if(isset($_GET['act']) and $_GET['act'] == ''){
                include_once 'newart1.php';
            }else if(isset($_GET['act']) and $_GET['act'] == 'see'){
                include_once 'showarticle1.php';
            }else if(isset($_GET['act']) and $_GET['act'] == 'pics'){
                include_once 'pics1.php';
            }else if(isset($_GET['act']) and $_GET['act'] == 'person'){
                include_once 'person.php';
            }else if(isset($_GET['act']) and $_GET['act'] == 'frd'){
                include_once 'frd1.php';
            }else if(isset($_GET['act']) and $_GET['act'] == 'modart'){
                include_once 'modart.php';
            }else{
                include_once 'newart1.php';
            }
        ?>
        </div>
…
```

11.4　我的文章

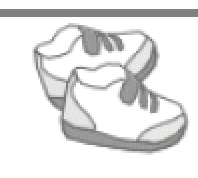
11.4.1　“我的文章”概述

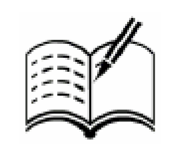
进入博客个人空间，或者单击博客个人空间中的“我的文章”超链接，将在内容显示区显示博客中已存在的文章的简要信息列表，其中包括文章标题、发布时间、部分文章内容和一个“阅读全文”超链接，如图 11.15 所示。



图 11.15　文章列表的运行效果

单击“阅读全文”超链接，打开 showarticle1.php 页面，将显示要查看的文章的全文和





Note

评论信息。同时，还会显示相关的一些操作，例如引用文章和发表评论。如果是本人的博客，登录后还可以修改文章、删除文章和删除评论，如图 11.16 所示。



图 11. 16　浏览文章页面的运行效果

11.4.2　“我的文章”功能实现过程

创建 newart1.php 文件，完成“我的文章”模块设计。首先设置分页信息，然后生成 SQL 查询语句，用来查找当前用户所发表过的文章，并返回查询结果。实现代码如下：

```
<?php
  include_once 'conn/conn.php';
  if(isset($_GET['uid'])){
    $num = 7;                                              //每页显示记录数
    if(!isset($_GET['curpage']) ){
        $curpage = 1;
    }else{
        $curpage = $_GET['curpage'];                       //当前页
    }
    $sql = "select * from tb_article where author = '".$_GET['uid']."' order by id desc ";
    $totnum = $conne->getRowsNum($sql);                    //记录数
    $totpage = ceil($totnum / $num);                       //页数
    $tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;  //SQL 查询语句
    $arr = $conne->getRowsArray($tmpsql);                  //获取数组
    $conne->close_rst();
?>
```

如果获取到的数组不为空，那么循环判断是否为引用，如果不是，则直接输出文章信息；如果是，那么根据引用的文章 ID 号来查找原文，并输出文章信息。最后，添加一个“阅读全文”超链接。代码如下：

```
if($arr != ''){
```






```
        foreach($arr as $value){
            if(!empty($value['artquote']) and !is_null($value['artquote'])){
                $tmparr = explode(',',$value['artquote']);
                $sql1 = "select * from tb_article where id = ".$tmparr[0];
                $artarr1 = $conne->getRowsArray($sql1);
                $conne->close_rst();
                …
            }else{
                …
            }
}
```

11.4.3　文章浏览功能实现过程

在文章浏览页（showarticle1.php）中，最核心的功能当然是浏览文章。首先，根据超链接传递的文章 ID，将文章信息全部显示出来。代码如下：

```
<?php
    include_once '../config.php';                                   //载入配置文件
    include_once 'conn/conn.php';                                   //载入数据库类
    $showartsql = "select * from tb_article where id = ".$_GET['artid']; //生成查询语句
    $artarr = $conne->getRowsArray($showartsql);                    //返回数组形式的记录
    $conne->close_rst();                                            //清空结果集
        if(!empty($artarr[0]['artquote']) and !is_null($artarr[0]['artquote'])){  //判断文章是否为引用
            $tmparr = explode(',',$artarr[0]['artquote']);          //得到引用的来源地址
            $sql1 = "select * from tb_article where id = ".$tmparr[0];//根据地址查询数据
            $artarr1 = $conne->getRowsArray($sql1);                 //返回结果集
            $conne->close_rst();
            …
            }else{
            …
            }
?>
```

然后，实现文章引用（其实质是引用文章地址）。当其他的博客用户看到好文章时，可以通过“引用”超链接，将文章 ID 保存到自己的文章列表中。通过对文章 ID 的访问，可以省去手工输入的麻烦，也在一定程度上节省了表空间。此外，如果博主想对自己的这篇文章进行宣传，也可以将引用地址复制下来，粘贴到其他地方来提高博客空间的人气。实现引用的代码如下：

```
[<a onclick="return artquote('<?php echo $_GET['uid'] ?>','<?php echo $_GET['artid']; ?>')" style="color:#FF6600;">引用</a>] 来自动添加到您的博客空间<br />
    <input id="copyadd" onclick="copyaddress()" type="text" size="50" value="<?php echo 'http://'.$_SERVER['HTTP_ HOST'].$_SERVER['REQUEST_URI']; ?>" readonly="readonly" style="border: 1px #FFFFFF solid;"  />
```

该段代码主要通过两个 JavaScript 自定义函数来控制。第一个函数 artquote()将博客用





Note

户、文章 ID 传给 showarticle_chk.php 页，根据返回结果进行处理。脚本文件 js\center.js 的代码如下：

```
//引用
function artquote(uid,quoteid){
        xmlhttp.open('get','showarticle_chk.php?act=quote&uid='+uid+'&quoteid='+quoteid,true);
        xmlhttp.onreadystatechange = function(){
            if(xmlhttp.readystate == 4 && xmlhttp.status == 200){
                msg = xmlhttp.responseText;
                if(msg == '1'){
                    alert('引用成功');
                }else if(msg == '2'){
                    alert('您还没有登录');
                }else if(msg == '3'){
                    alert('引用失败');
                }else if(msg == '4'){
                    alert('不允许引用自己的文章');
                }else{
                    alert(msg);
                }
            }
        }
        xmlhttp.send(null);
}
```

showarticle_chk.php 页处理和文章相关的操作，包括添加评论、删除评论和引用。其主要代码如下：

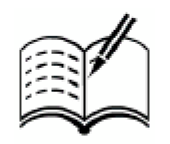
```
<?php
    session_start();                                            //开启 SESSION
    header('Content-Type:text/html;charset=gb2312');            //设置编码
    include_once 'conn/conn.php';                               //载入数据库类
    $artid = $_GET['artid'];                                    //文章 ID
    $quoteid = $_GET['quoteid'];                                //引用 ID
    $name = $_SESSION['name'];                                  //当前用户
    $uid = $_GET['uid'];                                        //原文章用户
    $act = $_GET['act'];                                        //何种操作
    $reback = '0';
    if(empty($act)){
      if($name != $uid){
        $selectsql = "select renum from tb_article where id = ".$artid;
        $renum = $conne->getFields($selectsql,0);
        $upsql = "update tb_article set renum = ".(++$renum)." where id = ".$artid;
        $num = $conne->uidRst($upsql);
      }
        $content = $_GET['cont'];
$sql = "insert into tb_review (artid,content,man) values('".$artid."','".$content."','".($name ==
"?'匿名':$name)."')";
        $num = $conne->uidRst($sql);
        $reback = '1';
```






```
    }else if($act == 'quote'){                          //如果是引用操作
        if(!empty($name) and !is_null($name)){
            if($_GET['uid'] == $name){                  //如果是引用了自己的文章
                $reback = '4';
            }else{
            $chkquote = $conne->getRowsNum("select artquote from tb_article where author = '".$name."'");
                if($chkquote >= 1){
                    $reback = '5';
                }else{
                    $selectsql = "select renum from tb_article where id = ".$quoteid;
                    $renum = $conne->getFields($selectsql,0);
                    $upsql = "update tb_article set renum = ".++$renum." where id = ".$quoteid;
                    $num = $conne->uidRst($upsql);
                    $oldid = $conne->getFields("select artquote from tb_article where id = ".$quoteid,0);
                    if(!empty($oldid) and !is_null($oldid)){
                        $saveid = $oldid.','.$quoteid;
                    }else{
                        $saveid = $quoteid;
                    }
        $sql = "insert into tb_article (typename,author,artquote) values('日记','".$name."','".$saveid."')";
                    $num = $conne->uidRst($sql);
                    if($num == 1){
                        $reback = '1';
                    }else{
                        $reback = '3';
                    }
                }
            }
        }else{
            $reback = '2';
        }
    }else if($act == 'delone'){
        $delonesql = "delete from tb_review where id= ".$_GET['rid'];
        $num = $conne->uidRst($delonesql);
        $reback = '1';
    }else if($act == 'delall'){
        $delallsql = "delete from tb_review where artid = ".$_GET['aid'];
        $num = $conne->uidRst($delallsql);
        $reback = '1';
    }
    echo $reback;
?>
```





Note

11.5 文 章 管 理

11.5.1 文章管理概述

对于一个博客系统来说，文章管理是最基本，也是最复杂的一个功能。在文章管理模块中提供了添加文章、文章列表、文章类别等功能，下列分别介绍。



图 11.17　添加文章页面的运行效果

11.5.2 添加文章功能实现过程

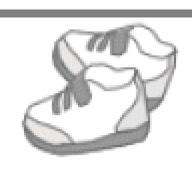
在添加文章页 addart.php 中创建一个表单，其中包括文章标题、文章类别、文字编辑区、文章内容等表单元素。该页面的运行效果如图 11.17 所示。页面中的部分表单元素如表 11.1 所示。

表 11.1　添加文章页面的主要元素属性及说明

元 素 名 称	元 素 属 性	说　　明
text	type="text" id="txt_title" size="68"	文章标题

select	<?php
	　　foreach($typearr as $value){
	?>
	　　<option value="<?php echo $value; ?>"><?php echo $value; ?> </option>
	<?php
	}
	?>
		文章类别

select	name="font" class="wenbenkuang" id="font" onChange="showfont (this.options[this.selectedIndex].value)"	文章字体
select	class="wenbenkuang" onChange="showsize(this.options[this.selectedIndex].value)"	字体大小
select	onChange="showcolor(this.options[this.selectedIndex].value)" name="color" size="1" class="wenbenkuang" id="select"	字体颜色
textarea	cols="75" rows="20" id="file" style="border:0px;width:520px;"	文章内容
button	id="enterbtn" onClick="return chkart()"	“提交”按钮



当用户填写完文章主题和文章内容后，单击“提交”按钮，将触发 onclick 事件，调用 chkart() 函数。该函数将文章标题、文章类别等相关消息进行验证后，传递给 addart_chk.php 页，根据 addart_chk.php 页的返回值进行不同的处理。chkart()函数存储于 js\article.js 脚本文件中，其关键代码如下。

```
function chkart(){
    if($('txt_title').value==""){                              //检测博客标题
        alert("博客标题名称不允许为空！");
        $('txt_title').focus();
        return false;
    }
    if($('file').value==""){                                   //检测文章内容
        alert("文章内容不允许为空！");
        $('file').focus();
        return false;
    }
    url = 'article/addart_chk.php?act=add&title='+$('txt_title').
value+"&arttype="+$('articletype'). value+"&cont=" +$('file').value;  //生成 URL
    xmlhttp.open('get',url,true);
        xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readystate == 4){
            if(xmlhttp.status == 200){
                msg = xmlhttp.responseText;                     //获取返回值
                if(msg == '1'){                                 //如果等于 1，发表成功
                    alert('文章发表成功');
                    xmlhttp.open('get','article/article.php',true);//刷新文章列表
                    xmlhttp.onreadystatechange = function(){
                        if(xmlhttp.readystate == 4 && xmlhttp.status == 200){
                            $('showmenu').innerHTML = xmlhttp.responseText;
                        }}
                    xmlhttp.send(null);
                }else if(msg == '2'){
                    alert('添加失败');
                }else{
                    alert(msg);
                }}}}
    xmlhttp.send(null);
}
```

addart_chk.php 处理页接到文章信息后，将新信息添加到数据库中，如果成功，返回 1。该页的代码如下：

```
<?php
session_start();                                            //开启 SESSION
header('content-type:text/html;charset=gb2312');            //设置编码
include "../Conn/conn.php";                                 //载入数据库类
$title=$_GET['title'];                                      //文章标题
$typename = $_GET['arttype'];                               //文章类别
$author=$_SESSION['name'];                                  //当前用户
```

```
$content=$_GET['cont'];                                         //文章内容
$reback = '0';
$sql="Insert Into tb_article (typename,title,content,author)
Values ('".$typename."','".$title."','".$content."','".  $author."')";     //insert 语句
$num = $conne->uidRst($sql);                                    //添加文章
if($num == 1){                                                  //如果添加成功
        $selectsql = "select upfile from tb_member where name = '".$author."'";
        $artnum = $conne->getFields($selectsql,0);              //得到上传文章数量
        $upsql = "update tb_member set upfile = ".(++$artnum)." where name = '".$author."'";
        $num = $conne->uidRst($upsql);                          //更新
        $reback = '1';
}else{
        $reback = '2';
}
echo $reback;
?>
```

11.5.3 文章列表功能实现过程

文章列表页列出了文章的类别、标题和点击率，并提供了全选、反选、删除和无刷新分页等功能，如图 11.18 所示。



图 11.18 文章列表页面的运行效果

文章列表页 article.php 的代码如下：

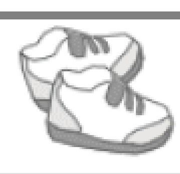
```
<?php
        session_start();                                        //开启 SESSION
        header('content-type:text/html;charset=gb2312');        //设置编码
        include_once '../conn/conn.php';
if(isset($_GET['act'])){                                        //判断 act 参数
        $act = $_GET['act'];
}else{
        $act="";
}
        $name = $_SESSION['name'];                              //获取当前用户
        $num = 10;                                              //每页显示记录数
if(isset($_GET['curpage'])){                                    //判断是否为当前页
        $curpage = $_GET['curpage'];
```



```
}else{
    $curpage ="";
}
    if($act == 'del'){                                             //如果是删除操作
        $rd = $_GET['rd'];                                         //得到要删除的 ID 序列
        $delsql = "delete from tb_article where id = -1 ";         //初始化删除语句
        if(!empty($rd)){                                           //如果删除序列不为空
            $delarr = explode(',',$rd);                            //拆分序列转为数组
            foreach($delarr as $value){                            //循环数组
                $delsql .= "or id = ".$value." ";                  //更新删除语句
            }
            $delnum = $conne->uidRst($delsql);                     //执行删除语句
        }
    }
        if(empty($curpage) or is_null($curpage)){                  //如果是第一次打开该页
            $curpage = 1;                                          //将当前页变量设为 1
        }
        if(empty($sql) or is_null($sql)){                          //如果 SQL 查询语句为空
    //创建新的查询语句
    $sql = "select id,typename,title,hitnum,renum from tb_article where author = '".$name."' order by id desc";
        }
        $totnum = $conne->getRowsNum($sql);                        //总记录数
        $totpage = ceil($totnum / $num);                           //总页数
        $tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;  //分页
        $arr = $conne->getRowsArray($tmpsql);                      //得到记录数组
?>
```

上述代码实现了删除数据及分页使用的功能，接下来就是文章列表的显示。代码如下：

```
<div id = 'showarticle'>
<table width="600" border="1" cellpadding="0" cellspacing="0" align="center">
    <tr>
        <td height="25" width="30" align="center" valign="middle"> </td>
        <td height="25" width="100" align="center" valign="middle">类别</td>
        <td height="25" width="370" align="center" valign="middle">标题</td>
        <td height="25" width="100" align="center" valign="middle">留言/点击率</td>
    </tr>
<?php
    foreach($arr as $key => $value){
?>
        <tr>
        <td align="center" valign="middle" height="25">
<input id="chk[<?php echo $key; ?>]" name="chk[]" type="checkbox" value="<?php echo $value['id']; ?>" /></td>
        <td align="center" valign="middle" height=25> <?php echo $value['typename']; ?></td>
        <td align="left" valign="middle" height=25> <?php echo $value ['title']; ?> </td>
    <td align="center" valign="middle" height=25> <?php echo $value['renum'].'/'.$value['hitnum']; ?> </td></tr>
```





Note

```
<?php
}
```

以下是全选、反选、删除及分页的实现，其中删除是全选、反选操作的延续。代码如下：

```
if($totnum > 0){                                    //如果有记录
    ?>
        <tr>
            <td colspan="4" height="30" align="right" valign="middle">
                <a href="#" onclick="return alldel('<?php echo count($arr); ?>')">全选</a>
                <a href="#" onclick="return overdel('<?php echo count($arr) ?>')">反选</a>
<!--删除操作调用 del()函数，并传递了两个参数-->
<button id="delbtn" name="delbtn" class="btn" onclick="return del('<?php echo $_SERVER['SCRIPT_NAME']; ?>','
<?php echo count($arr); ?>',<?php echo $curpage; ?>)">删除选择</button>
<!-- 翻页超链接，将调用 pagination()函数 -->
            <?php echo ($curpage == 1)?"首页":"
<a onclick='return pagination(\"".$_SERVER['SCRIPT_NAME']." \",1)'>首页</a>"; ?>
            <?php echo ($curpage==1)?"上一页":"
<a onclick='return pagination(\"".$_SERVER ['SCRIPT_NAME']. "\",".($curpage-1).")'>上一页</a>"; ?>
            <?php echo ($curpage == $totpage)?'下一页':"
<a onclick='return pagination(\"".$_SERVER ['SCRIPT_NAME']."\",".($curpage+1).")'>下一页</a>"; ?>
            <?php echo ($curpage==$totpage)?"尾页":"
<a onclick='return pagination(\"".$_SERVER['SCRIPT_NAME']."\",".($totpage).")'>尾页</a>"; ?>
        当前是第<?php echo $curpage; ?>页/共<?php echo $totpage; ?>页<?php echo $totnum; ?>条记录
<!-- 跳转操作，调用 jumpmem()函数 -->
跳 转 到 ： <select id="jump" name="jump" onchange="return jumpmem('<?php echo $_SERVER['SCRIPT_NAME']; ?>')">
            <?php
                for($i=1;$i<=$totpage;$i++){
                    if($i == $curpage)
                        echo "<option value=".$i." selected>".$i."</option>";
                    else
                        echo "<option value=".$i.">".$i."</option>";
                }
            ?>
            </select> 页</td>
        </tr>
    <?php
        }
?>
</table>
</div>
```

删除、翻页和跳转操作分别调用了脚本文件 choose.js 中的 3 个函数。删除函数 del()



包含 3 个参数，这 3 个参数分别代表删除处理页的 URL、当前页的记录总数和当前页的页码。

函数中首先声明了一个数组 rd，然后循环判断当前页中的复选框状态，将要删除的复选框的 vlaue 值保存到 rd 中；接着将 rd 和当前页的页码传给处理页；最后调用 rerst()函数将处理页的输出结果显示到指定区域。choose.js 文件的删除代码如下：

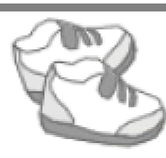
```
//删除所选
function del(chkurl,tot,curpage){
	var rd=new Array();                                    //声明数组
	for(i=0,j=0;i<tot;i++){                                //循环判断复选框
		if(!$('chk['+i+']').checked){
		}else{                                             //如果复选框被选中
			rd[j] = $('chk['+i+']').value;                 //将 value 值保存到 rd 数组中
			j++;
		}
	}
	if(rd == ''){                                          //如果没有删除，提示错误
		alert('请选取要删除数据!');
		return false;
	}
	url = chkurl+"?act=del&rd="+rd+'&curpage='+curpage;   //生成 URL
	xmlhttp.open('get',url,true);                          //创建新请求
	xmlhttp.onreadystatechange = rerst;                    //调用 rerst()函数
	xmlhttp.send(null);
	return false;
}
function rerst(){
	if(xmlhttp.readystate == 4){
		if(xmlhttp.status == 200){                         //判断请求是否完成
		$('showarticle').innerHTML = xmlhttp.responseText; //将返回值显示到指定区域
		}
	}
}
```

对翻页的处理统一使用 pagination()函数。该函数有两个参数：一个是处理页的 URL，另一个是当前页的页码。代码如下：

```
//分页
function pagination(chkurl,curpage){
	url = chkurl+'?act=next&curpage='+curpage;            //生成 URL
	xmlhttp.open('get',url,true);                          //创建新请求
	xmlhttp.onreadystatechange = rerst;                    //调用 rerst()函数
	xmlhttp.send(null);
	return false;
}
```

跳转是翻页的另一种形式。翻页只能一页一页地翻，而跳转可以直接跳到指定的页面。代码如下：

```
//跳转
```





Note

```
function jumpmem(chkurl){
        jumppage = $('jump').value;                          //获取跳转页数
        url = chkurl+'?curpage='+jumppage;                   //生成 URL
        xmlhttp.open('get',url,true);                        //创建请求
        xmlhttp.onreadystatechange = rerst;                  //调用 rsrst()函数
        xmlhttp.send(null);
        return false;
}
```

11.5.4　文章类别功能实现过程

在文章类别模块中可以实现文章类别的添加、删除。文章类别页面的运行效果如图 11.19 所示。



图 11.19　文章类别页面的运行效果

文章的类别数据被保存到用户信息表的一个类别字段中。默认有一个“日记”类别，当有多个类别时，将使用逗号分割。显示文章类别时，首先获取到该用户信息中类别字段的值，将字符串使用逗号分割成数组，最后由 foreach 循环显示所有类别。文章类别处理文件 arttype.php 的代码如下：

```
<?php
        session_start();                                                    //开启 SESSION
        header('content-type:text/html;charset=gb2312');                    //设置编码
        include_once '../conn/conn.php';                                    //载入数据库类文件
        $act = $_GET['act'];                                                //获取 act 值
        $name = $_SESSION['name'];                                          //获取用户名
        $sql = "select arttype from tb_member where name = '".$name."'";    //生成查询语句
        $typefields = $conne->getFields($sql,0);                            //获取字段值
?>
<table id="addtypefm" width="300" border="1" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td colspan="2" height="25" align="center" valign="middle">文章类别管理</td>
    </tr>
<?php
        $typearr = explode(',',$typefields);                    //使用逗号分隔字符串存为数组
        foreach($typearr as $value){                            //循环输出数组
```



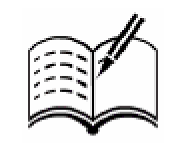
Note

```
?>
    <tr>
        <td width="200" height="25" align="center" valign="middle"> <?php echo $value; ?></td>
        <td height="25" align="center" valign="middle"> 
<button id="deltype" value="<?php echo $value; ?>" class="btn"
onclick="return delarttype('<?php echo $typefields; ?>','<?php echo $value;?>')">删除</button></td>
    </tr>
<?php
    }
?>
    <tr>
        <td width="200" height="25" align="center" valign="middle">
<input id="addarttype" name="addarttype" type="text" class="txt" /></td>
<td height="25" align="center" valign="middle"><button id="addtype" value="<?php echo $value; ?>" class="btn"
onclick="return addarttype('<?php echo $typefields; ?>')">添加新类</button></td>
    </tr>
</table>
```

如要添加文章类别，单击“添加新类”按钮后，将调用 addarttype()函数，并将原类别字段传给该函数。addarttype()函数将创建一个新请求，并根据请求的输出值进行不同的处理。addarttype()函数存储于 js\article.js 脚本文件中，其代码如下：

```
//添加文章类别
function addarttype(typestr){
    if($('addarttype').value == ''){                    //判断类别名称是否为空
        alert('请输入添加类别名称');
        $('addarttype').focus();
        return false;
    }
    url = 'article/arttype_chk.php?act=add&typestr='+typestr+'&typename='+$('addarttype').value ;
    xmlhttp.open('get',url,true);                        //创建新请求
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readystate == 4){
            if(xmlhttp.status == 200){
                msg = xmlhttp.responseText;              //获取处理页输出结果
            //根据返回的不同结果进行不同的处理
                if(msg == '2'){
                    alert('添加失败');
                }else if(msg == '3'){
                    alert('添加名称重复');
                }else if(msg == '1'){
                    showarttype();
                }else{
                    alert(msg);
                }
```

```
                        }
                }
        }
        xmlhttp.send(null);
}
```

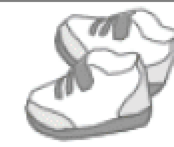


arttype_chk.php 是 addarttype()函数调用的处理页，将获取到的原类别字段后面加上逗号和新类别名称，再将该字段保存。其代码如下：

```
<?php
        session_start();                                        //开启 SESSION 支持
        header('Content-Type:text/html;charset=gb2312');        //设置编码
        include_once '../../conn/conn.php';                     //载入数据库类
        $act = $_GET['act'];                                    //获取 act 值
        $typestr = $_GET['typestr'];                            //原类别字符串
        $typename = $_GET['typename'];                          //要获取的类别字符串
        $arr = explode(',',$typestr);                           //将原字符串转换为数组
        $reback = '0';
        $name = $_SESSION['name'];                              //获得当前用户
        if($act == 'add'){                                      //如果是添加操作
                if(in_array($typename,$arr)){                   //判断名称是否重复
                        $reback = '3';
                }else{
                        $typestr .= ','.$typename;              //将新类别名称添加到原类别后面
                        $sql = "update tb_member set arttype = '".$typestr."' where name = 
'".$name."'";
                        $num = $conne->uidRst($sql);            //执行更新语句
                        if($num == 1){
                                $reback = '1';
                        }else{
                                $reback = '2';
                        }s
                }
        }
…
```

删除文章类别和添加文章类别大同小异，但要注意两点。（1）不允许删除默认的“日记”类别，如果删除的是“日记”类别，则弹出警告；（2）将删除的类别下的所有文章，全部放到默认的“日记”目录下。删除文章类别操作是在 arttype_chk.php 文件中完成的，其代码如下：

```
//接添加类别代码
else if($act == 'del'){
                if($typename == '日记'){
                        $reback = '3';
                }else{
                        $tmpstr = '日记';
                        for($i=1;$i<count($arr);$i++){
                                if($arr[$i]  != $typename){
```



```
                    $tmpstr .= ','.$arr[$i];
                }
            }
    $changsql = "update tb_article set typename = '日记'where author = '".$name."' and typename=
'". $typename."'";
            $num = $conne->uidRst($changsql);
            $sql = "update tb_member set arttype = '".$tmpstr."' where name = '".$name."'";
            $num = $conne->uidRst($sql);
            if($num == 1){
                $reback = '1';
            }else{
                $reback = '2';
            }
        }
    }
    echo $reback;
```

11.6　用户管理

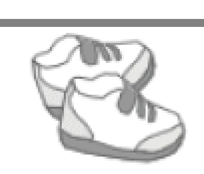
11.6.1　用户管理概述

博客后台管理系统主要由六大模块构成，分别是用户管理模块、文章管理模块、图片管理模块、管理员设置模块、系统日志模块以及数据备份模块。本节重点讲解用户管理模块，其运行效果如图 11.20 所示。



图 11.20　后台用户管理模块运行效果

顾名思义，用户管理模块就是用来管理注册用户的。在该模块中提供了多项功能，如查询博客、推荐博客、冻结博客、查看详细资料和删除博客等。其中推荐博客和冻结博客的实现方法基本相同，而查看详细资料和删除博客在前面也接触过，这里只讲解查询博客和推荐博客的实现方法。



11.6.2　用户管理功能实现过程

首先来看一下用户管理模块的首页 member.php。在 member.php 文件中，将数据库中存储的用户信息进行分页输出，并且完成各种附加功能的设置。其运行效果如图 11.21 所示。



	编号	用户账号	email	真实姓名	性别	博客点击率	推荐	冻结/解冻	详细资料
	25	leonsk			男	0	否	冻结	详细资料
	24	mrlx			男	0	否	冻结	详细资料
	23	lx111			男	0	否	冻结	详细资料
	22	a还珠格格	778452@qq.com	王佳丽	女	3	否	冻结	详细资料
	21	博客皇帝	1111111QQ@。com	张其卫	男	6	推荐	冻结	详细资料
	20	my BLOG	732565048QQ@.com	小丽	男	5	否	冻结	详细资料
	19	I LOVE YOU	845129549qq@.COM	123	女	5	否	冻结	详细资料

全选 反选 删除选择　　首页 上一页 下一页 尾页 当前是第1页/共1页7条记录 跳转到: 1 页

图 11.21　用户管理页面的运行效果

member.php 页的代码如下：

```
<?php
	session_start();						//开启 SESSION
	include_once '../conn/conn.php';				//载入数据库类
?>
<script language="javascript" src="../js/choose.js"></script>	//载入 JS 文件
<?php
$num = 10;							//每页显示记录数
if(empty($_GET['act'])){						//获取当前是第几页
	$act = "";
}else{
		$act = $_GET['act'];					//当前页
	}
		$sql = "";						//SQL 语句
	if(empty($curpage) or is_null($curpage)){			//获取当前是第几页
		$curpage = 1;
	}else{
		$curpage = $_GET['curpage'];				//当前页
	}
	if(empty($fields) or is_null($fields)){			//排列顺序，默认为 id
		$fields = 'id';
	}else{
		$fields = $_GET['fields'];				//依照该字段排列记录
	}
	 //是否有查询内容，如果没有
	if(!isset($_REQUEST['cont'])){
	 //直接生成 SQL 查询语句，使用$fields 字段作为排序依据
$sql = "select id,name,email,realname,sex,birthday,hitnum,freeze,isnominate from tb_member order by ".$fields." desc";
		//如果有查询内容，就获取要查询的字段、查询方式和查询内容
	}else{
		$querymem = $_REQUEST['querymem'];
		$signslt = $_REQUEST['signslt'];
		$cont = $_REQUEST['cont'];
```

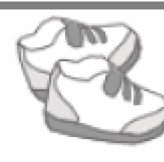



```
			//初始化变量$sql 为一个无条件的查询语句
		$sql = 'select id,name,email,realname,sex,birthday,hitnum,freeze,isnominate from tb_member ';
			//根据查询字段所在的组，生成不同的 where 子句
		if(in_array($querymem,array('id','upfile','uppics','hitnum','birthday'))){
			$sql .= " where ".$querymem." ".$signslt." ".$cont;
		}else if(in_array($querymem,array('name','realname','qq','email'))){
			if($signslt == 'exac'){
				$sql .= "where ".$querymem." = '".$cont."'";
			}else if($signslt == 'mist'){
				$sql .= "where ".$querymem." like '%".$cont."%'";
			}
		}
			//添加 order by 子句
		$sql .= " order by ".$fields." desc";
	}
	$totnum = $conne->getRowsNum($sql);				//记录数
	$totpage = ceil($totnum / $num);				//页数
	$tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;
	$arr = $conne->getRowsArray($tmpsql);
	?>
```

11.6.3　查询博客功能实现过程

查询博客是根据博客不同的信息来查询特定的博客，如 ID 号、点击率等。查询页是单独的一页，被包含在 member.php 页中。当用户查询时，该页将查询字段、查询方式和查询内容传给 member.php 页，再由 member.php 页进行处理（member.php 页的处理过程在上面已经介绍过）。查询页 query.php 的代码如下：

```
<?php
	include_once '../conn/conn.php';			//载入数据库类
?>
<script language="javascript" src="../js/query.js"></script>
<table border="0" cellpadding="0" cellspacing="0" align="center">
<form id="queryfm" name="queryfm" method="post" action="member.php?act=query" onSubmit="return chkquery()">
	<tr>
		<td width="50" height="25">查询：</td>
		<td width="100">
		<select id="querymem" name="querymem" onChange="return changequery()">
			<option value="id">编号</option>
			<option value="name">账号</option>
			<option value="realname">真实姓名</option>
			<option value="email">email</option>
			<option value="qq">qq</option>
			<option value="birthday">出生日期</option>
			<option value="upfile">文章数</option>
			<option value="uppics">上传图片</option>
```


Note

```
                <option value="hitnum">点击率</option>
            </select>
            </td>
            <td width="50" align="center" valign="middle"><div id="sign">
            <select id="signslt" name="signslt">
                <Option value="&gt;">大于</Option>
                <option value="=">等于</option>
                <option value="&lt;">小于</option>
            </select>
            </div></td>
            <td width="100"><div id="input"><input id="cont" name="cont" type="text" size="30" /></div></td>
            <td><input id="querybt" name="querybt" type="submit" value="查询"></td>
        </tr>
</form>
</table>
```

在查询页中，当用户选择不同的查询条件时，查询方式也会随之发生变化。例如查询 ID、点击率时，查询条件有大于、等于和小于；当查询账号、真实姓名、E-mail 时，查询条件变为精确和模糊。这是通过 query.js 脚本来控制的。实现代码如下：

```
function changequery(){
    key = document.getElementById('querymem').value;            //获取查询条件
    switch(key){
        /*当查询条件和数字有关系时*/
        case 'id':
        case 'upfile':
        case 'uppics':
        case 'hitnum':
            document.getElementById('sign').innerHTML = '<select id="signslt" name="signslt">
<Option value= "&gt;">大于</Option><option value="=">等于</option><option value="&lt;">小于</option></select>';
            document.getElementById('cont').value='';
            break;
        /*当查询条件和字符串有关系时*/
        case 'name':
        case 'realname':
        case 'email':
        case 'qq':
        case 'title':
        case 'author':
        case 'typename':
        case 'upname':
        case 'upauthor':
        case 'operator':
            document.getElementById('sign').innerHTML = '<select id="signslt" name="signslt">
<Option value= "exac">精确</Option><option value="mist">模糊</option></select>';
```




```
                document.getElementById('cont').value='';
                break;
            /*当查询条件和时间有关系时*/
            case 'birthday':
            case 'firsttime':
            case 'uptime':
                document.getElementById('sign').innerHTML = '<select id="signslt" name="signslt">
<Option value= "&gt;">大于</Option><option value="=">等于</option><option value="&lt;">小于</option></select>';
                document.getElementById('cont').value='查询格式：YYYY-MM-DD';
                break;
            /*省略部分为其他模块的查询方式设置*/
            …
        }
        document.getElementById('cont').focus();
        document.getElementById('cont').select();
    }
```

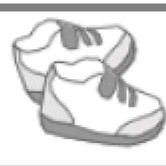
11.6.4 推荐博客功能实现过程

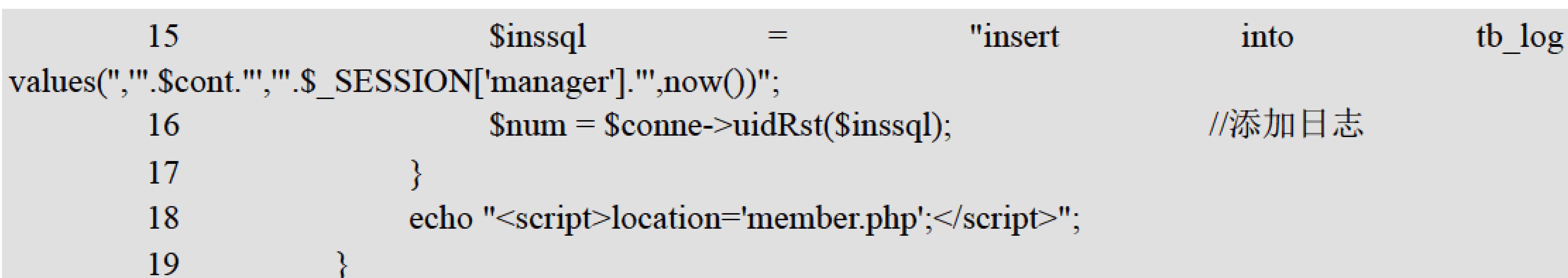
当用户为推荐博客时，该用户的博客空间将在首页的显著位置显示，这样可以提高网站和个人空间的人气和访问量。推荐博客实现方法很简单，在用户表中设置一个字段，名称是 isnominate，类型为 int。当用户注册时，字段的默认值为 0；当设为推荐博客时，将该字段设为 1。该操作从 member.php 文件中设置的超链接开始，其代码如下。

```
…
<a href="member_chk.php?act=nominate&id=<?php echo $value['id']; ?>&isnominate=<?php echo $value ['isnominate']; ?>"><?php echo ($value['isnominate']==0?"否":"推荐"); ?></a>
…
```

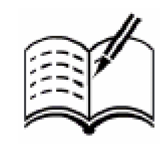
链接到 member_chk.php 文件，根据博客用户的 ID 号和 isnominate 字段的值来更新用户表。其代码如下：

```
<?php
    session_start();
    header('Content-Type:text/html;charset=gb2312');
    include_once '../conn/conn.php';
    $act = $_GET['act'];
    if($act == 'del'){
        …                                                   //删除操作
    }else if($act == 'nominate'){                           //推荐博客
        $id = $_GET['id'];                                  //获取用户 ID 号
        $nominate = ($_GET['isnominate']+1) % 2;            //将获取的字段值加 1 再模 2
        $sql = "update tb_member set isnominate = ".$nominate." where id = ".$id;
        $cont = '推荐用户操作，账号：'.$id;                   //系统日志内容
        $num = $conne->uidRst($sql);                        //更新字段值
        if(!empty($num)){
```





```
                    $inssql = "insert into tb_log values('','".$cont."','".$_SESSION['manager']."',now())";
                    $num = $conne->uidRst($inssql);                 //添加日志
               }
               echo "<script>location='member.php';</script>";
          }
```



Note

11.7 技 术 提 炼

11.7.1 文件包含技术

通过文件包含技术实现文件的包含，有助于网站页面的规划、管理和更新。例如，网站的头文件、尾文件和数据库的连接文件，都可以通过包含的方式进行载入，这样就避免了在每个页面中编写相同的代码，减少了代码的冗余，同时也便于对这些内容的修改和更新。

在博客模块的个人首页中通过 include_once()函数包含文件。include_once()函数应用于脚本执行期间同一个文件有可能被包含超过一次的情况下，以确保它只被包含一次，从而避免出现函数重定义、变量重新赋值等问题。也就是说，使用 include_once()函数包含文件时，如果该文件中的代码已经被包含，则不会被再次包含。

下面是博客个人首页的左侧包含文件代码：

```
<!--左侧-->
     <div id="leftfloatcont">
        <div id="personinfo">
           <?php include_once 'personinfo.php'; ?>
        </div>
        <div id="date">
           <?php include_once 'date.html'; ?>
        </div>
        <div id="showmessdiv">
           <?php include_once 'mess1.php'; ?>
        </div>
     </div>
```

11.7.2 SQL 查询技术

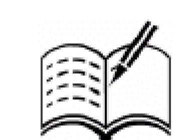
在“我的文章”模块中，总体是采用 SQL 语句查询技术，从数据库中根据查询条件完成数据的循环输出。其中用到的 SQL 技术如下：

1. 排序功能

排序分为升序和降序，升序是从小到大，而降序则相反，是从大到小。排序主要有两个关键字：升序 asc、降序 desc。其语法格式如下：

```
select * from 表名 order by 字段名 asc/desc
```





Note

2. 查询记录总数

将数据库中的记录总数查询出来，使用的关键字是 count 和 as。其语法格式如下：

```
select count (*) as (自定义字段) from  表名
```

3. 条件查询

按照指定的一个或多个条件进行查询，使用的关键字是 where。其语法格式如下：

```
select * from  表名  where  字段=条件
```

下面讲解一下这些 SQL 查询技术在本模块中的具体应用。

（1）“我的文章”中应用的查询语句，其代码如下：

```
$sql = "select * from tb_article where author = '".$_GET['uid']."' order by id desc ";
$totnum = $conne->getRowsNum($sql);
$totpage = ceil($totnum / $num);
$tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;
$arr = $conne->getRowsArray($tmpsql);
$conne->close_rst();
```

查询文章表（tb_article）中的所有记录，查询条件是 author 的值与超链接传递的值相同，并以 ID 为条件进行降幂排列。最后输出查询结果，完成我的文章标题、发表时间、部分文章内容的循环输出。

（2）文章浏览中应用的查询语句，其代码如下：

```
$showartsql = "select * from tb_article where id = ".$_GET['artid'];   //生成查询语句
$artarr = $conne->getRowsArray($showartsql);                         //返回数组形式的记录
$conne->close_rst();                                                  //清空结果集
```

以超链接中传递的文章的 ID 为条件，执行查询操作；查询的结果就是有关这条记录的详细信息，即文章的详细内容。

11.7.3　会话处理技术

想要使用文章管理模块，前提是用户必须登录，匿名用户是无法访问这些功能的；想要删除文章和评论，前提是当前用户要么是管理员，要么是文章拥有者，否则不会显示删除功能。这两方面的控制都需要 SESSION 的配合，本节就来讲一下 SESSION 的应用及常见的问题处理。

SESSION 的中文含义为“会话”，是指用户从进入网站开始，直到关闭网站这段时间内，所有网页共同使用的公共变量的存储机制。SESSION 比 Cookie 更有优势：SESSION 是存储在服务器端的，不易被伪造；SESSION 的存储没有长度限制；SESSION 的控制更容易等。

PHP 主要是通过会话（SESSION）处理函数来对 SESSION 进行控制和使用。常用的处理函数如表 11.2 所示。


Note

表 11.2　PHP 常用的会话处理函数

函　　数	功 能 说 明
session_start();	开启SESSION或返回已经存在的SESSION
$_SESSION['name'] = value;	注册一个SESSION变量
session_id()	设定或取得当前的session_id值
isset($_SESSION['name'])	检测指定的SESSION值是否存在。Isset不仅可以检测SESSION，还可以检测其他类型，如isset($_post['name'])、isset($_get['name'])等
session_regenerate_id()	更改session_id的值
session_name()	返回或改变当前SESSION的name。如果要改变当前SESSION的name值，必须在session()之前调用session_name()函数，而且session_name不能全部是数字，否则会不停地生成新的session_id
unset($_SESSION['name'])	删除名为name的SESSION。注意，不能写成unset($_SESSION)，否则会禁止整个会话的功能
session_destroy()	结束当前会话，删除所有session

下面介绍几个使用 SESSION 时要注意的问题。

1. 尽可能地将 session_start 放到第 1 行

这种情况是新手最容易犯的错误。产生的错误代码为：

```
Warning: session_start() [function.session-start]: Cannot send session cache limiter - headers already sent…
```

其原因就是在使用 session_start()之前，有 HTML 代码输出。也许有的读者会说：session_start 之前没有任何代码，绝对没有，但还是出现了这个错误。此时应检查程序，看看是不是有空行或类似 echo 语句的输出。如果有，则去掉，因为就算是一个小小的空格都是不可以的。为了避免这类错误的发生，应尽可能地将 session_start 放到第 1 行。

2. 在使用 SESSION 之前一定要先写 session_start()

大多数读者在使用 SESSION 之前都能先调用 session_start()函数，但对于 session_destroy()函数却经常忽略。session_destroy()虽然是结束当前会话并删除所有 SESSION，但在删除之前，也要先开启 SESSION 支持才可以，不然会产生如下的错误代码。

```
session_destroy() [function.session-destroy]: Trying to destroy uninitialized session in…
```

所以，凡是在使用 SESSION 函数的页面中，都要加上 session_start()。

3. 删除所有 SESSION

如果想删除所有 SESSION，但又不想结束当前会话，用 unset()一个一个删除实在是太麻烦了，最简单的办法就是将一个空数组赋给$_SESSION，如$_SESSION = array()，这样就解决了。

11.7.4　冻结、解冻技术

所谓用户冻结技术，就是将用户冻结后，他将无法登录。有冻自然就有解，如果解冻，



用户便可以正常登录。用户冻结技术主要是针对一些不守网站规则的用户而设立。其实用户冻结技术与用户推荐、文章审核等技术的实现原理是相同的。

采用数据库条件式方法，在指定数据库内设置单独的字段作为冻结技术的条件，当条件为真时用户便是正常运行（解冻），如果条件为假用户便停止运行（冻结），这就是冻结和解冻的原理。本模块中具体的冻结、解冻操作是在 admin\member\member.php 文件中完成的，其代码如下：

```
$id = $_GET['id'];
        $freeze = ($_GET['fz']+1) % 2;
        $sql = "update tb_member set freeze = ".$freeze." where id = ".$id;
        $cont = '冻结(解冻)操作，账号：'.$id;
        $num = $conne->uidRst($sql);
        if(!empty($num)){
            $inssql = "insert into tb_log values('','".$cont."','".$_SESSION['manager']."',now())";
            $num = $conne->uidRst($inssql);
```

11.7.5 配置文件

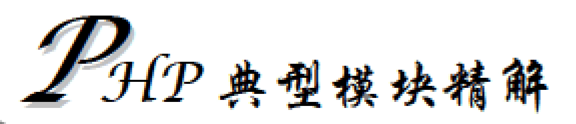
在配置文件中设置了服务器的目录、博客目录、上传图片目录、图片大小上限等信息。配置文件 config.php 存储于前台。配置文件代码如下：

```
<?php
    define('PATH',$_SERVER['DOCUMENT_ROOT']);         //服务器目录
    define('ROOT','/db_blog1.0 /');                   //博客目录
    define('ADMIN','admin/');                         //后台目录
    define('PIC','center/pics/image/');               //上传图片目录
    define('BAK','sqlbak/');                          //备份目录
    define('HEADGIF','headgif/');                     //头像目录
    define('MYSQLPATH','D:\\xampp\\MySQL\\bin\\');    //MySQL 执行文件路径
    define('MYSQLHOST','127.0.0.1');                  //MySQL 服务器 IP
    define('MYSQLDATA','db_blog');                    //MySQL 数据库
    define('MYSQLUSER','root');                       //MySQL 账号
    define('MYSQLPWD','111');                         //MySQL 密码
    define('MAXSIZEPIC',500000);                      //允许上传的图片的最大字节数
?>
```

11.7.6 公共函数

博客系统的公共函数有两个用途：一是返回指定文件夹下的文件列表；二是判断上传文件的后缀。公共函数存储于 center\inc\func.php，其代码如下：

```
<?php
//返回文件夹下的文件列表
function show_file($f_name){
     $d_open = opendir($f_name);                      //打开指定目录
     $num = 0;
     while($file = readdir($d_open)){                 //循环输出目录下文件
```








Note

```
            $filename[$num] = $file;                     //将文件保存成数组
            $num++;
        }
        closedir($d_open);                               //关闭目录
        return $filename;                                //返回文件数组
}
/*
 *判断文件后缀
 *$f_type：允许的文件后缀类型
 *$f_upfiles：上传文件名
 */
function f_postfix($f_type,$f_upfiles){
        $is_pass = false;
        $tmp_upfiles = split("\.",$f_upfiles);            //将后缀字符串分给数组
        $tmp_num = count($tmp_upfiles);                   //获取数组大小
        if(in_array(strtolower($tmp_upfiles[$tmp_num - 1]),$f_type))      //上传的后缀是否
在允许范围内
            $is_pass = $tmp_upfiles[$tmp_num - 1];       //保存后缀
        return $is_pass;                                  //返回结果
}
?>
```

第12章

在线编程词典模块

（自学视频、源程序：配套资源\mr\12\）

“在线编程词典”是明日科技编程词典软件的网页版，实现了编程词典软件基本的浏览功能，其目的是让浏览者通过网页体会编程词典软件的海量数据资源给编程带来的极大便利。

通过本章的学习，读者将体会到一种全新的编程方式，不是单纯地为某个企业开发一个网站，而是将企业开发的某个软件以网页的形式展示给大家。在这种全新的背景下，学习程序开发的技术、技巧，积累开发经验。

- 如何设计树状导航菜单
- 如何向 Flash 中传递参数
- 如何用 Smarty 注册模板函数——读取文件
- 了解 PDO 数据库抽象层
- 超链接中的图片切换技术



12.1 在线编程词典概述



Note

12.1.1 模块概述

编程词典是由吉林省明日科技有限公司研发的一款编程辅助软件，是明日科技数年开发经验、图书编撰经验的积累。

编程词典系列软件以其特有的学习模式、全面的编程资源（技术资源库、视频资源库、实例资源库、项目资源库、界面资源库、源码资源库、工具资源库等）、高效的查询与智能检索功能，向读者诠释了一种全新的学习方法。

无论您是初学编程的爱好者、大中专院校的师生、初中级程序设计人员，还是具有编程经验的开发人员或者从事软件研发的专家，使用编程词典系列软件都可以让您轻松上手，快速编程，高效开发。为了让更多的编程爱好者了解编程词典软件，亲身体会它给编程带来的便利，下面就来开发一个在线编程词典。

12.1.2 系统功能结构

“在线编程词典”是明日科技编程词典软件的网页版，实现了编程词典软件基本的浏览功能，其目的是让浏览者通过网页体会编程词典软件的海量数据资源给编程带来的极大便利。

在线编程词典具备以下功能：

- ☑ 编程词典内容的分类、分块输出。
- ☑ 便捷的树状导航菜单。
- ☑ 输出内容的权限控制。
- ☑ 体验内容的合理输出。

在线编程词典网站的系统功能结构如图 12.1 所示。

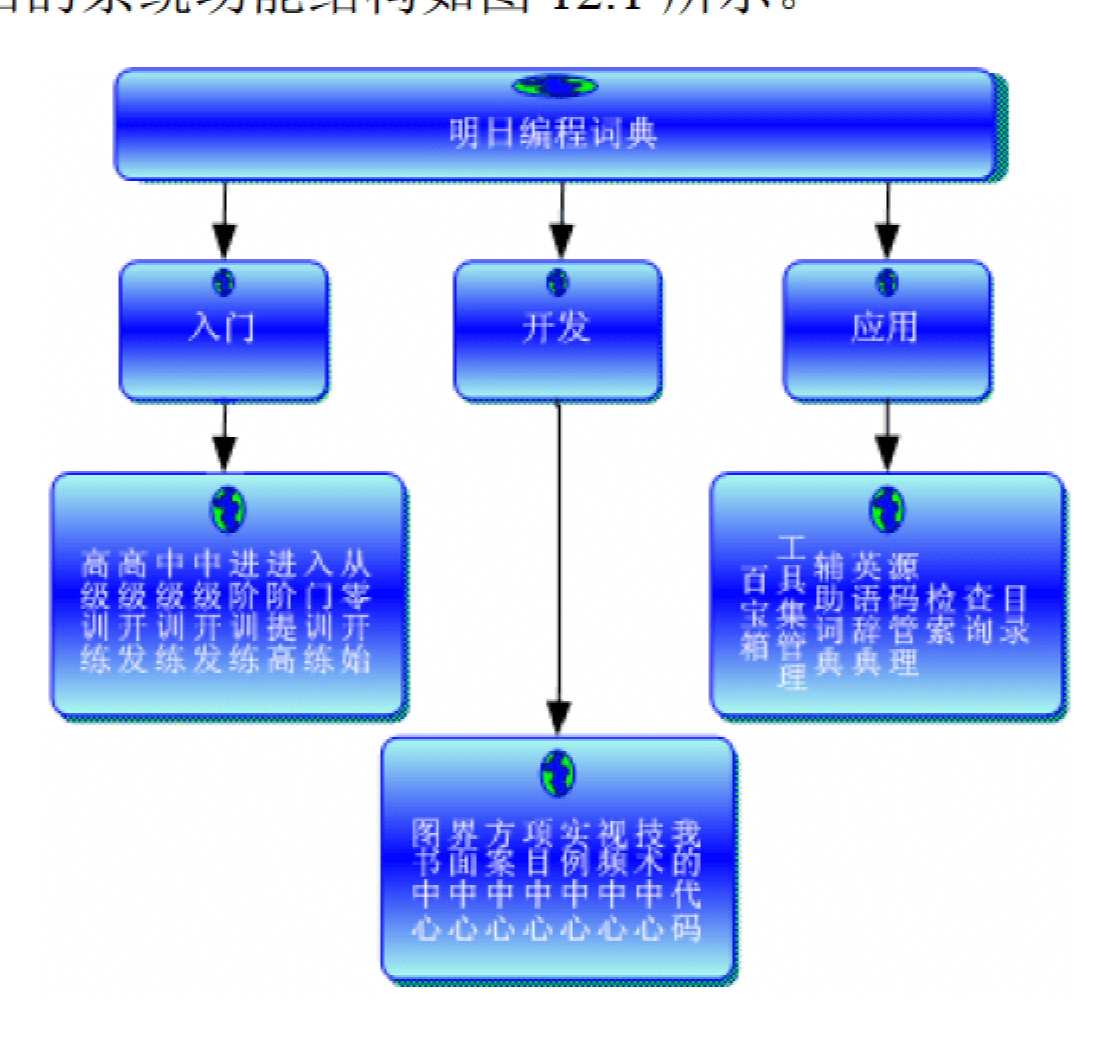


图 12.1　在线编程词典网站的功能结构图



Note

12.1.3　文件夹架构

在线编程词典网站的文件夹目录结构如图 12.2 所示，其中包括了本模块中所有的文件以及文件夹。



图 12.2　在线编程词典网站的文件夹架构

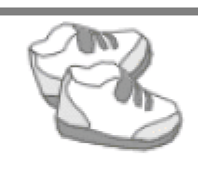
12.1.4　程序预览

为了让读者对本系统有个初步的了解和认识，下面给出本系统的几个页面运行效果图，如果想查看完整的效果图，参见配套资源源程序。

编程词典在线体验版入门模式中“从零开始”的运行效果如图 12.3 所示。



图 12.3　从零开始



Note

编程词典在线体验版开发模式中“技术中心”的运行效果如图 12.4 所示。

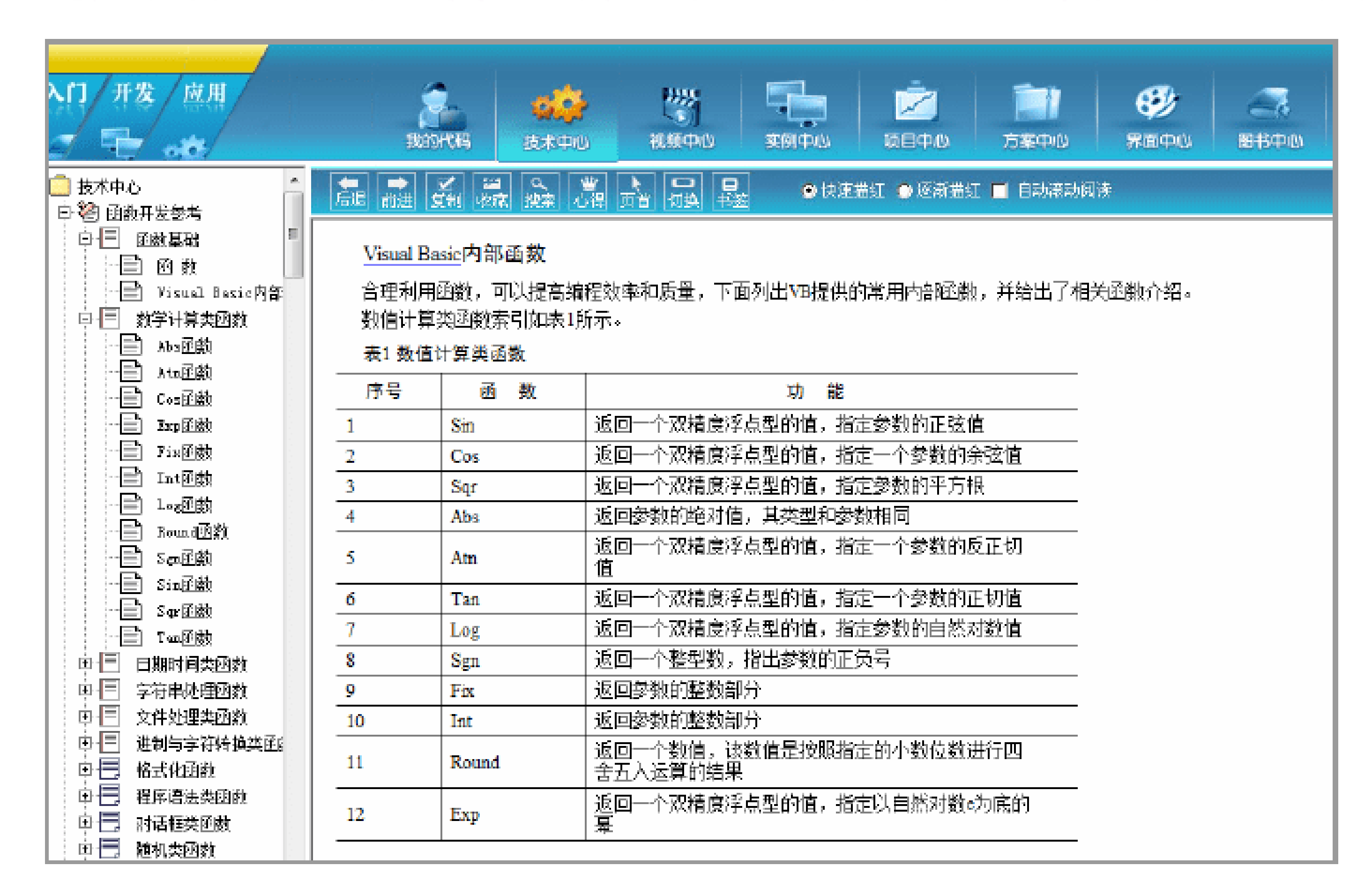


图 12.4　技术中心

编程词典在线体验版应用模式中“目录”的运行效果如图 12.5 所示。

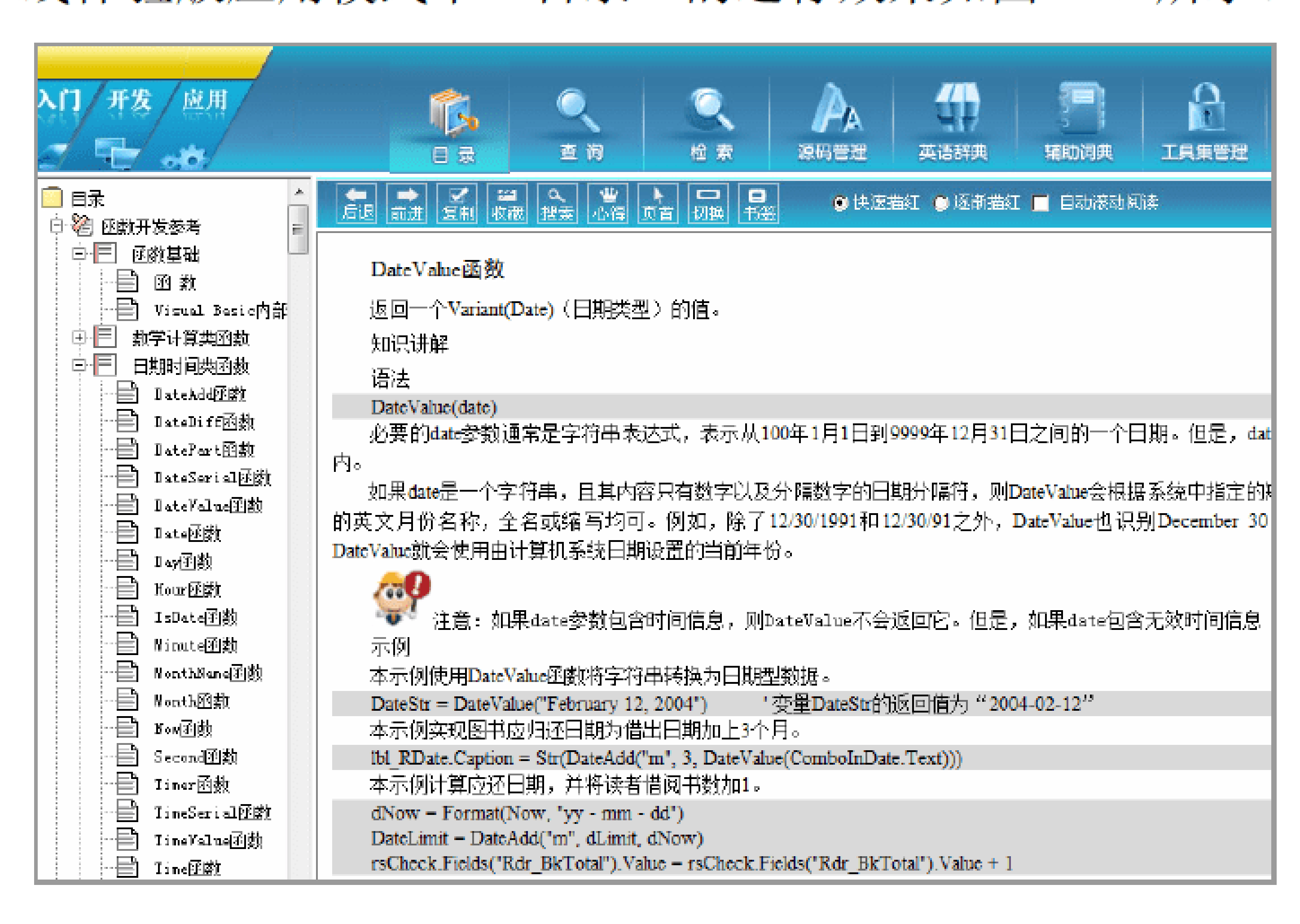


图 12.5　目录

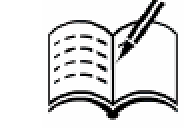
12.2　数据库设计

虽然 PHP 的最佳搭档是 MySQL 数据库，但是为了丰富读者数据库方面的知识，这里使用 MS SQL Server 2000 作为本程序的数据库。

12.2.1　创建数据库

在操作系统的任务栏中单击“开始”菜单，选择“所有程序”/Microsoft SQL Server/“企业管理器”命令，启动企业管理器，如图 12.6 所示。





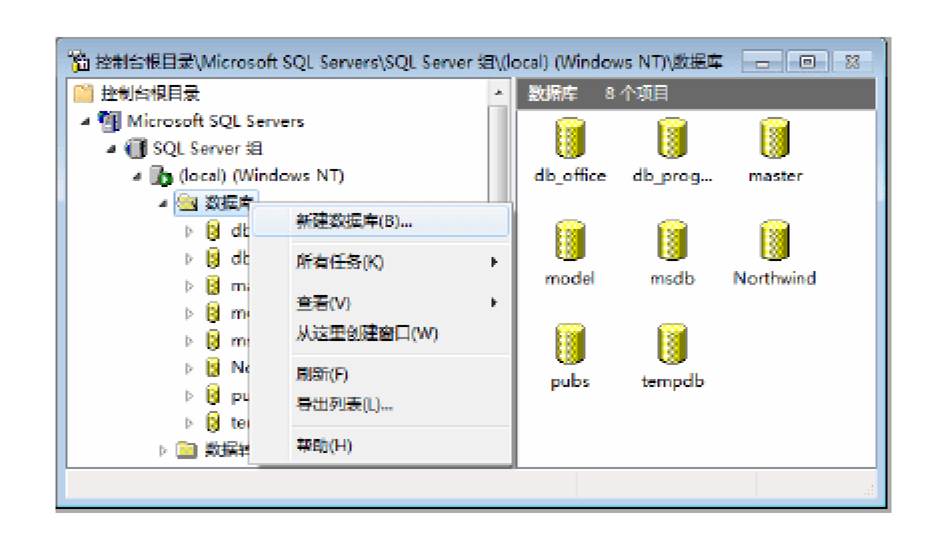


图 12.6 企业管理器

在图 12.6 所示的窗口中，右击“数据库”选项，在弹出的快捷菜单中选择“新建数据库”命令， 打开“数据库属性”对话框，如图 12.7 所示。在该对话框中输入数据库的名称，然后单击“确定”按钮，数据库创建成功。

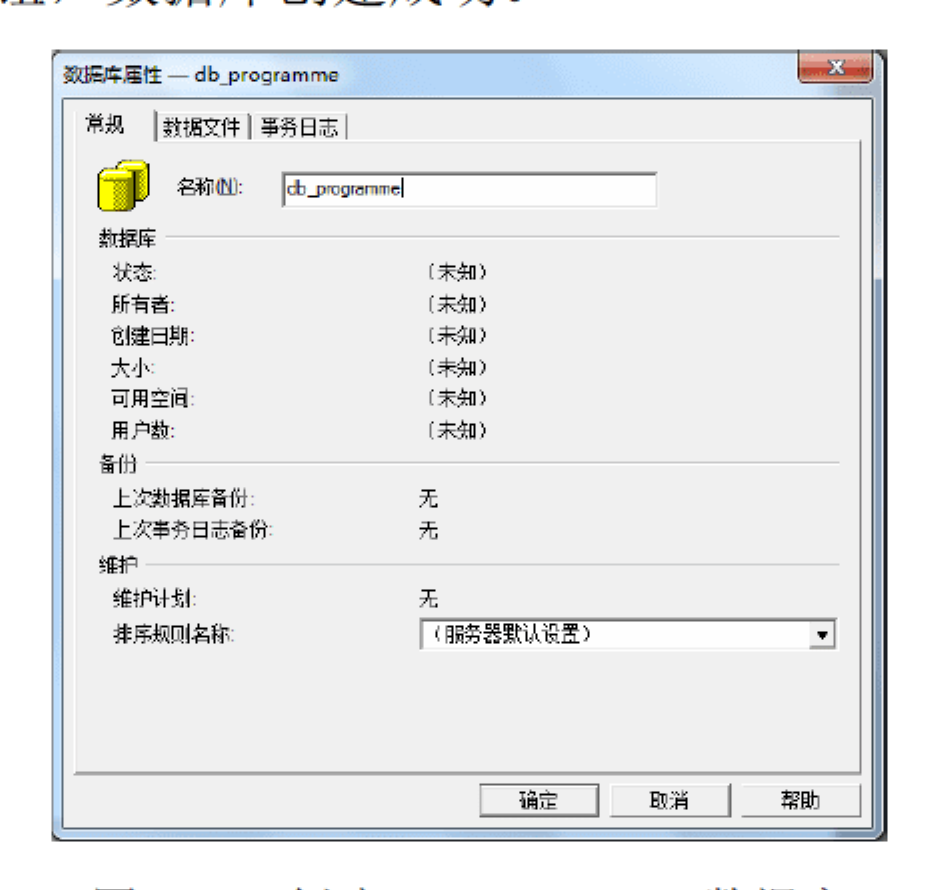


图 12.7 创建 db_programme 数据库

12.2.2 创建数据表

创建表的过程就是定义表列的过程。这里在 db_programme 数据库中创建一张名为 study 的数据表，操作步骤如下。

（1）启动企业管理器，展开服务器组及指定的服务器。

（2）展开“数据库”节点，展开指定的数据库 db_programme，右击“表”选项，在弹出的快捷菜单中选择“新建表”命令或直接在工具栏中单击✳按钮。

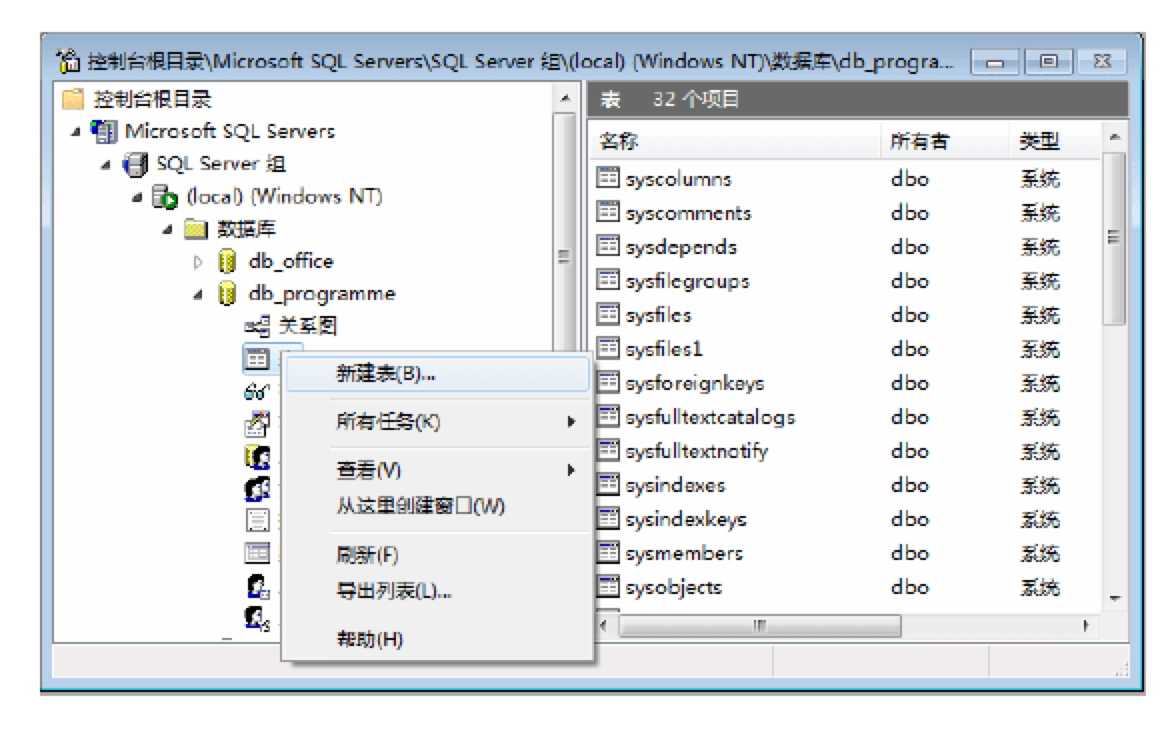
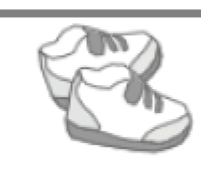


图 12.8 创建表



（3）在弹出的如图 12.9 所示的窗口中，输入表的字段名，同时设置表的数据类型、长度和允许空等属性。



Note

提示：

如果在“允许空”栏中设置了不允许为空，则插入或修改记录时此选项不允许为空值，否则会弹出错误提示。

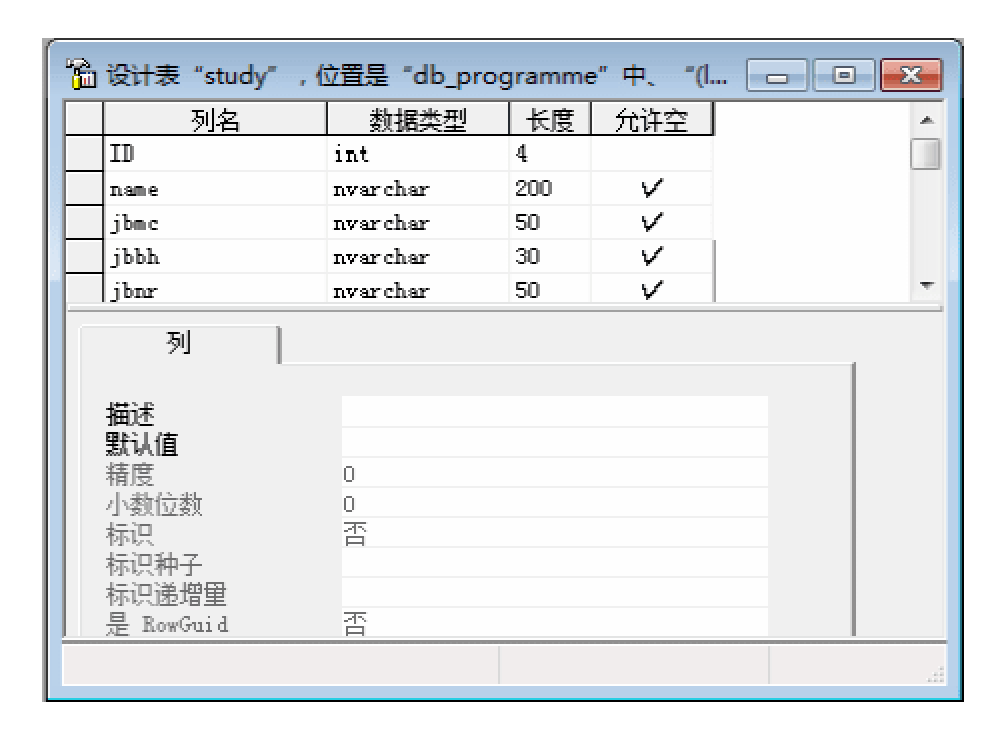


图 12.9　编辑表字段

（4）插入或删除列时，可以右击相应的列，在弹出的快捷菜单中选择“插入列”或“删除列”命令即可。

（5）将 ID 设置为主键。选中列名为 ID 的记录，单击鼠标右键，在弹出的快捷菜单中选择“设置主键”命令，或者直接单击工具栏上的按钮，即可将该列设置为主键，如图 12.10 所示。

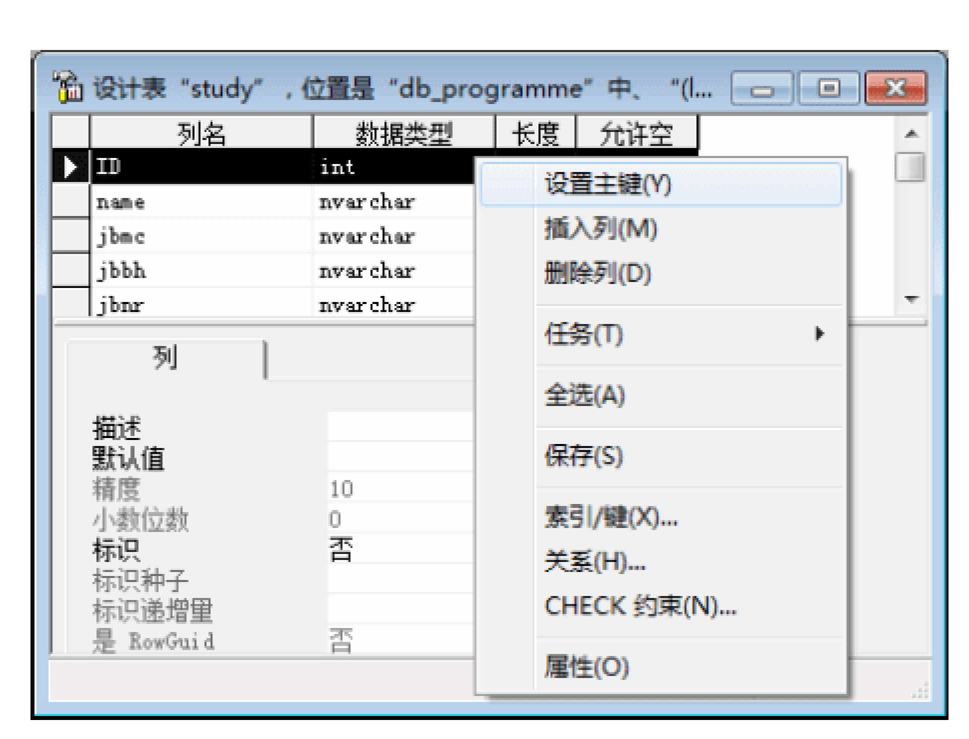
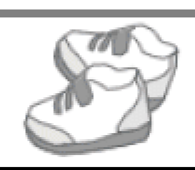


图 12.10　设置主键

（6）设置完成后，单击工具栏中的“保存”按钮，在弹出的如图 12.11 所示对话框中输入新建表的名称“study”，单击“确定”按钮，即可完成数据表的创建。



图 12.11　“选择名称”对话框



说明：

这里介绍的只是在 SQL Server 服务器下创建数据库、数据表的方法，并没有对本程序中所使用的数据库、数据表进行详细的讲解。如果要运行本程序或者想了解本程序所使用的数据库，必须从配套资源中复制 db_programme 数据库，并且要附加到本机的 SQL Server 2000 数据库中。



12.2.3　分离、附加数据库

在 SQL Server 2000 数据库中，可以通过“分离数据库”来实现对数据库文件的复制。分离后的数据库文件在服务器中不能使用，必须重新附加后才可以使用。

1.　分离数据库

这里以 db_programme 数据库为例，讲解分离数据库的操作步骤。

（1）启动企业管理器，展开服务器组及指定的服务器。

（2）展开“数据库”节点，右击 db_programme 数据库，在弹出的快捷菜单中选择“所有任务”/“分离数据库”命令，如图 12.12 所示。

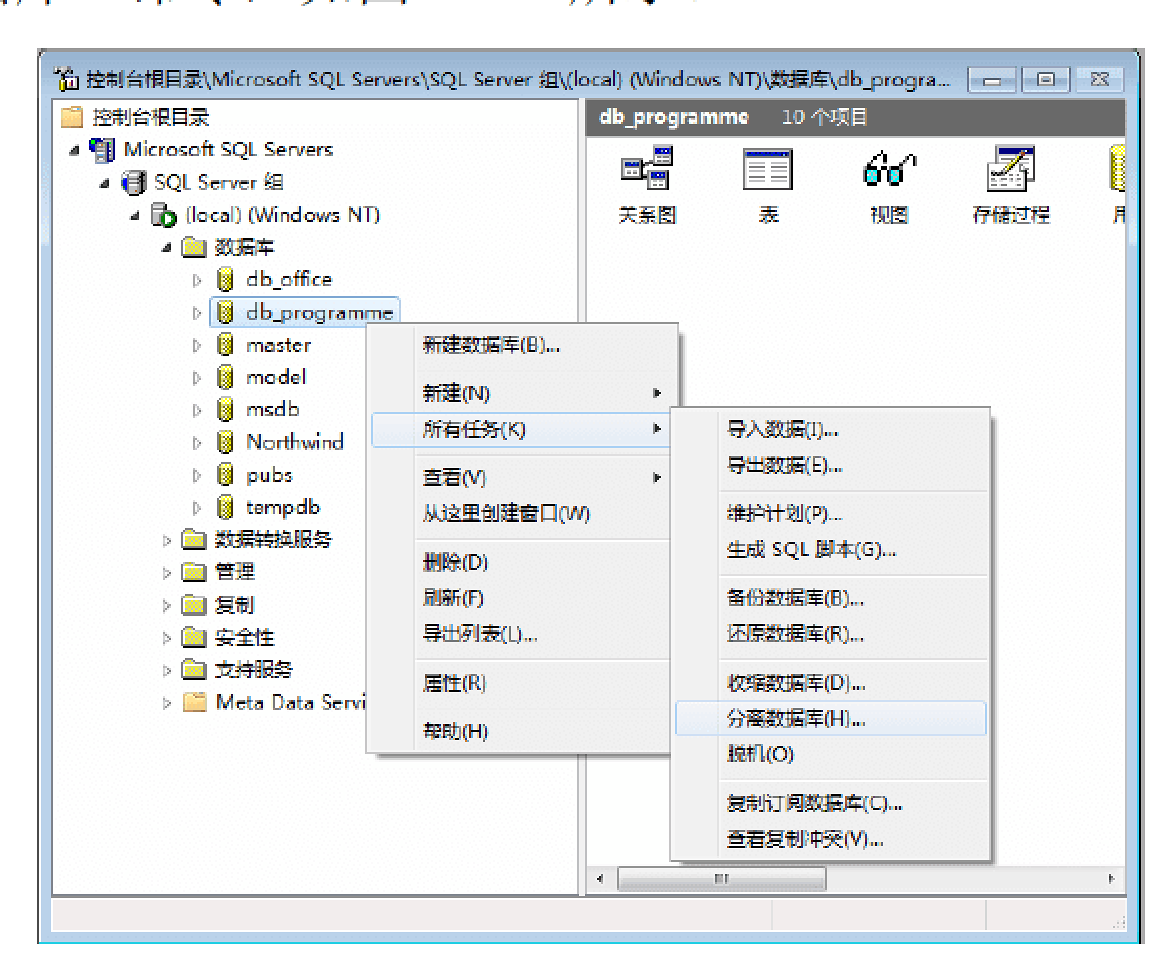


图 12.12　分离数据库

（3）在弹出的对话框中单击“确定”按钮，最后弹出数据库分离成功的提示信息。

2.　附加数据库

附加数据库对应分离数据库，由它完成分离后的数据库文件的附加操作。在本书的配套资源中提供的就是可以供读者附加的 db_programme 数据库文件。附加数据库的操作步骤如下：

（1）启动企业管理器，展开服务器组及指定的服务器。

（2）右击“数据库”节点，在弹出的快捷菜单中选择“所有任务”/“附加数据库”命令，如图 12.13 所示。

（3）在弹出的对话框中选择数据库文件，确定附加的数据库，然后单击“确定”按钮，即可完成分离后的数据库文件的附加操作，如图 12.14 所示。





Note

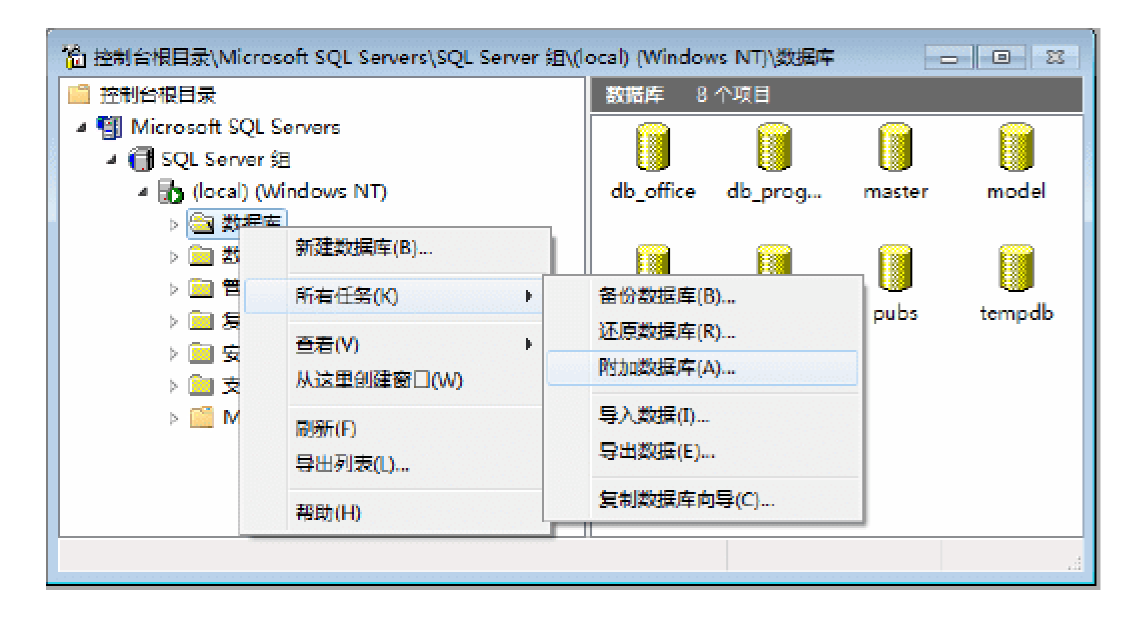


图 12.13　选择“所有任务”/“附加数据库”命令

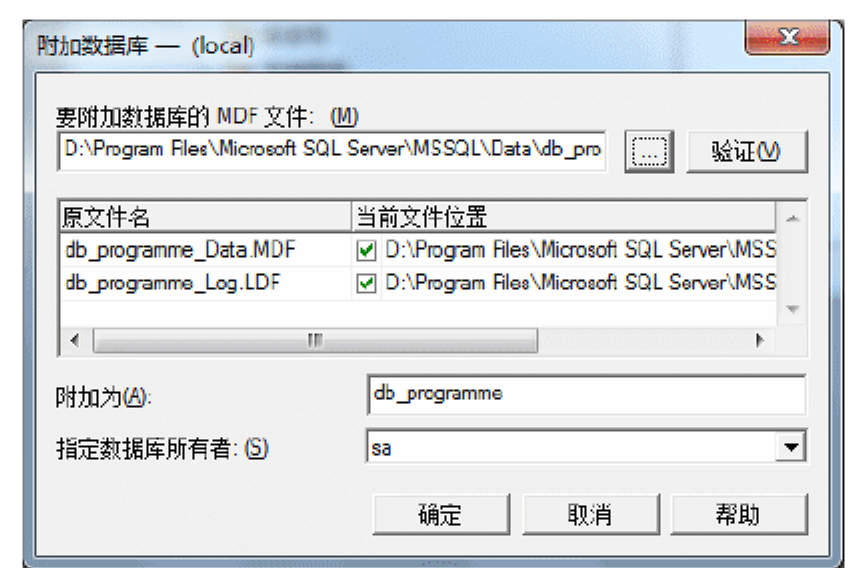


图 12.14　附加数据库

12.3　首 页 设 计

首页的内容应突出网站主题及特色，这在网站的生命周期中起着非常重要的作用。在线编程词典网站首页应用框架设计，模拟编程词典软件的页面布局。

12.3.1　首页概述

在线编程词典网站首页的设计效果如图 12.15 所示，其中包括网站模块导航、功能导航条、内容树状导航、内容展示区和滚动导航条。

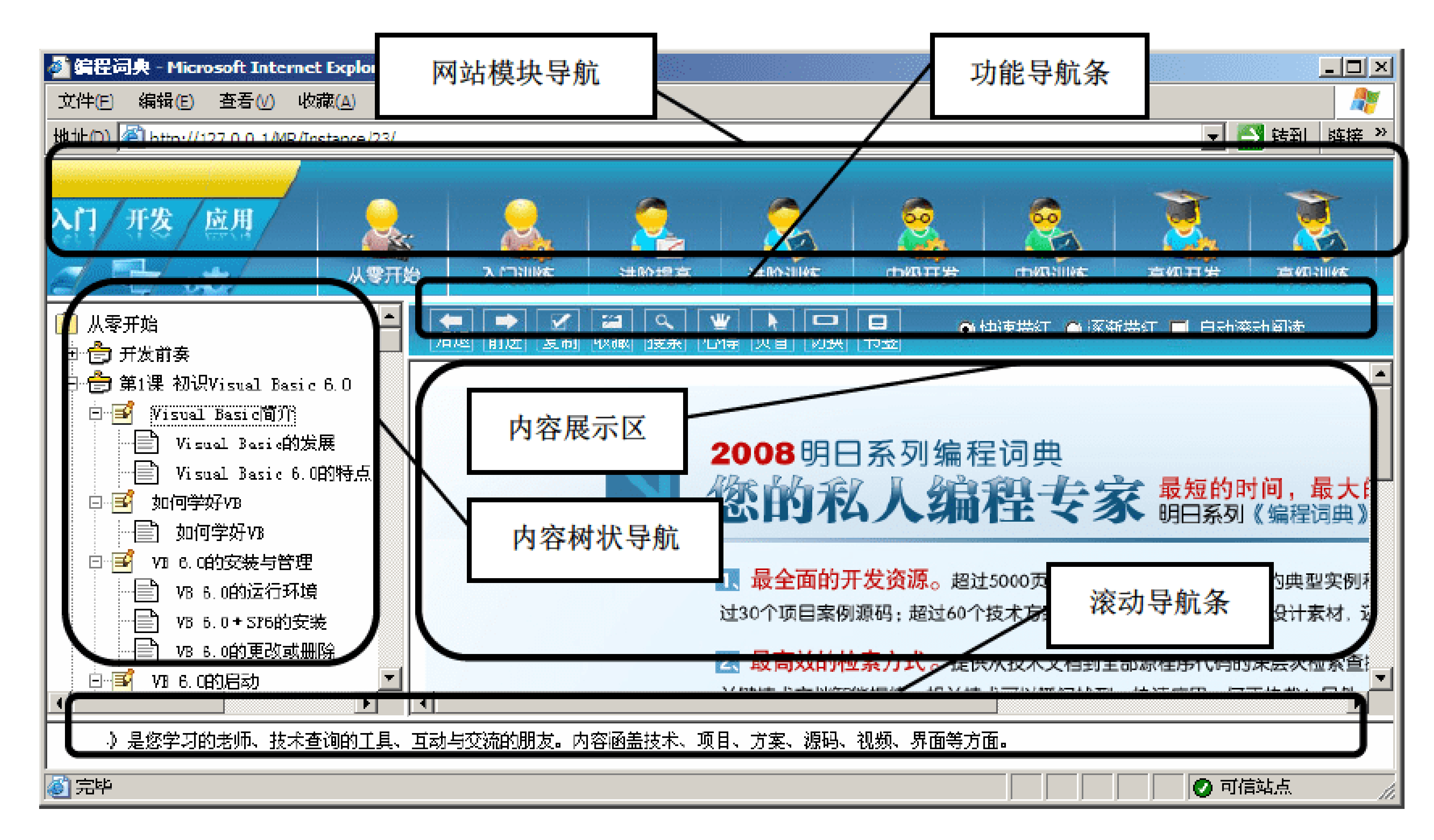


图 12.15　在线编程词典首页

12.3.2　首页的实现过程

在线编程词典网站的首页由两个文件组成，即模板页 index.html 和动态文件 index.php。在 index.html 模板页中，应用框架技术设计网页的布局。其关键代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
```





Note

```
<title>{$conn_type}编程词典</title>
</head>
<link rel="stylesheet" type="text/css" href="css/link.css">
<frameset rows="93,*,25" frameborder="0">
<frame src="pd_top.php?conn_type={$conn_type}" name="topwindow" scrolling="no">
    <frameset cols="200,*">
        <frame src="pd_left.php?conn_type={$conn_type}" name="leftwindow" frameborder="1">
        <frameset rows="37,*" frameborder="0">
          <frame src="pd_title.php" name="titlewindow" scrolling="no">
              <frame src="pd_main.php" name="mainwindow" scrolling="auto" frameborder="1">
        </frameset>
    </frameset>
 <frame src="pd_bottom.php" name="bottomwindow" scrolling="no" frameborder="1">
</frameset><noframes></noframes>
```

框架中的文件分配如下。

☑ 框架主体文件：index.html、index.php。

☑ 网站模块导航：pd_top.html、pd_top.php。

☑ 内容树状导航：pd_left.html、pd_left.php。

☑ 功能导航条：pd_title.html、pd_title.php。

☑ 内容展示区：pd_main.html、pd_main.php。

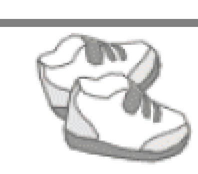
☑ 滚动导航条：pd_bottom.html、pd_bottom.php。

在 index.php 文件中，首先包含 Smarty 配置文件，然后创建模板变量，最后指定模板页。其代码如下：

```
<?php
header("content-type:text/html; charset=gb2312");                //设置编码格式
require_once("system/system.inc.php");                           //调用配置文件
if(isset($_GET['conn_type'])){
      $conn_type=$_GET['conn_type'];
}else{
      $conn_type="从零开始";
}
$smarty->assign('conn_type',$conn_type);                         //设置标题模板变量
$smarty->display('index.html');                                  //指定模板页
?>
```

12.4　网站模块导航设计

网站模块导航是编程词典软件功能模块的展示区域，它会根据入门、开发和应用 3 个类别，分别展示出每个类别中包括的功能。



12.4.1 网站模块导航概述

网站模块导航根据网站的 LOG 中传递的参数进行跳转，如果 LOG 中传递的是入门的数据，那么网站模块导航中将输出入门类中的功能模块，如图 12.16 所示。



图 12.16　入门类中的功能模块展示

如果 LOG 中传递的是开发的数据，那么网站模块导航中将输出开发类中的功能模块，如图 12.17 所示。



图 12.17　开发类中的功能模块展示

如果 LOG 中传递的是应用的数据，那么网站模块导航中将输出应用类中的功能模块，如图 12.18 所示。



图 12.18　应用类中的功能模块展示

12.4.2 网站模块导航实现过程

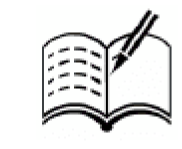
网站模块导航的实现，主要应用到 pd_top.html、pd_top.php，以及 js 文件夹下的 flash1.js 和 top.js 文件。

pd_top.php 是动态 PHP 文件，其主要功能就是定义模板变量传递参数，以及指定模板页。其代码如下：

```
<?php
session_start();                                   //初始化 SESSION 变量
require_once 'system/system.inc.php';              //包含配置文件，数据库连接、操作文件
if(isset($_GET['conn_type'])){
        $conn_type=$_GET['conn_type'];
}else{
        $conn_type="从零开始";
}
if(isset($_GET['conn_id'])){
        $conn_id=$_GET['conn_id'];
}else{
        $conn_id="20510111";
}
$smarty->assign('conn_type',$conn_type);          //定义模板变量传递参数
$smarty->assign('conn_id',$conn_id);              //定义模板变量传递参数
```

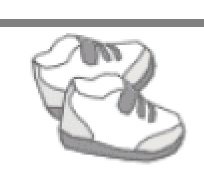


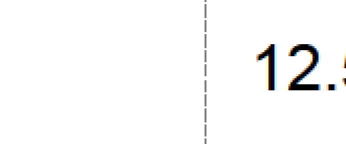
```
$smarty->display('pd_top.html');                    //指定模板页
?>
```



pd_top.html 是模板页。首先，通过 script 标签调用 js 脚本文件夹下的 top.js 和 flash1.js 文件。然后，定义 JavaScript 脚本，执行 flash1 方法，向 Flash 中传递参数值。最后，根据模板变量$conn_id 的值进行判断，进而输出不同类别的功能模块；同时为输出的功能模块创建超链接，链接到 pd_left.php 文件，同时传递相应的参数。其关键代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>{$conn_type}编程词典</title>
</head>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
❶<script type="text/javascript" src="js/top.js"></script>
❷<script src="js/flash1.js"></script>
❸<body onLoad="MM_preloadImages('images/criterion/images002.jpg')">
<div style="padding-top:inherit; width:100%; height: 93px; background:url (images/images_013.jpg);">
<div style="padding-top:inherit; width:171px; height: 93px; float:left; ">
        <script type="text/javascript">
❹                flash1('{$conn_type}');
        </script>
</div>
<div id="conn" style=" padding-top:0px; width:80%; height: 93px; float:right;">
❺       {if $conn_id=='20510111' or $conn_id==''}
        <div id="accidence" style="padding-top:0px; width: 100%; height: 93px; ">
                <ul>
❻               <li style="display:inline; width: 89px; height: 93px; float: center; text-align:center; font-size: 12px">
<a href="pd_left.php?conn_id=20510111&table_name=study&class_type=从零开始" target="leftwindow">
<img src="images/accidence/images_01.jpg" border="0" id="Image1"
onMouseOver="MM_swapImage('Image1','','images/accidence/images_001.jpg',1)"
onMouseOut="MM_swapImgRestore()" /></a></li>
                <li style="display:inline; width: 94px; height: 93px; float: center; text-align:center; font-size: 12px">
<a href="pd_left.php?conn_id=20610111&table_name=works&class_type=入门训练" target="leftwindow">
<img src="images/accidence/images_02.jpg" border="0" id="Image2"
onMouseOver="MM_swapImage('Image2','','images/accidence/images_002.jpg',1)"
onMouseOut="MM_swapImgRestore()" /></a></li>
                </ul>
        </div>
        {/if}
…<!--省略了部分代码-->
</div>
</div>
</body>
</html>
```











关键代码解析

❶ top.js：导入的 JS 脚本文件，控制超链接图片的样式。

❷ flash1.js：向 Flash 中传递参数使用的方法。

❸ MM_preloadImages：通过 onLoad 事件调用 top.js 中定义的方法。

❹ flash1()：执行 flash1.js 脚本中定义的方法，并传递参数{$conn_type}。

❺ {if $conn_id=='20510111' or $conn_id==''}：通过 Smarty 模板中的 if 语句判断网站模块属于哪一类。

❻ a：定义超链接，链接到 pd_left.php 文件，并传递相应的参数值，同时应用 onMouseOver 和 onMouseOut 事件调用 JavaScript 脚本中的方法，控制超链接图片的样式。

12.5 内容树状导航设计

在内容树状导航中，通过树状导航菜单来展示指定功能模块中包含的内容。

12.5.1 内容树状导航概述

在内容树状导航中，根据网站模块导航中传递的参数值进行判断，进而通过树状导航菜单输出对应功能模块中包含的内容。

在内容树状导航中不仅要包含 pd_left.html 和 pd_left.php 两个文件，还要根据网站模块导航中传递的参数值来调用入门类（accidence.html、accidence.php）、开发类（developer.html、developer.php）或者应用类（criterion.html、criterion.php）的内容。

如果是入门类，内容树状导航中默认展示的是“从零开始”功能模块的内容，如图 12.19 所示。



图 12.19　入门类的内容树状导航

如果是开发类，内容树状导航中默认展示的是“技术中心”功能模块的内容，如图 12.20 所示。

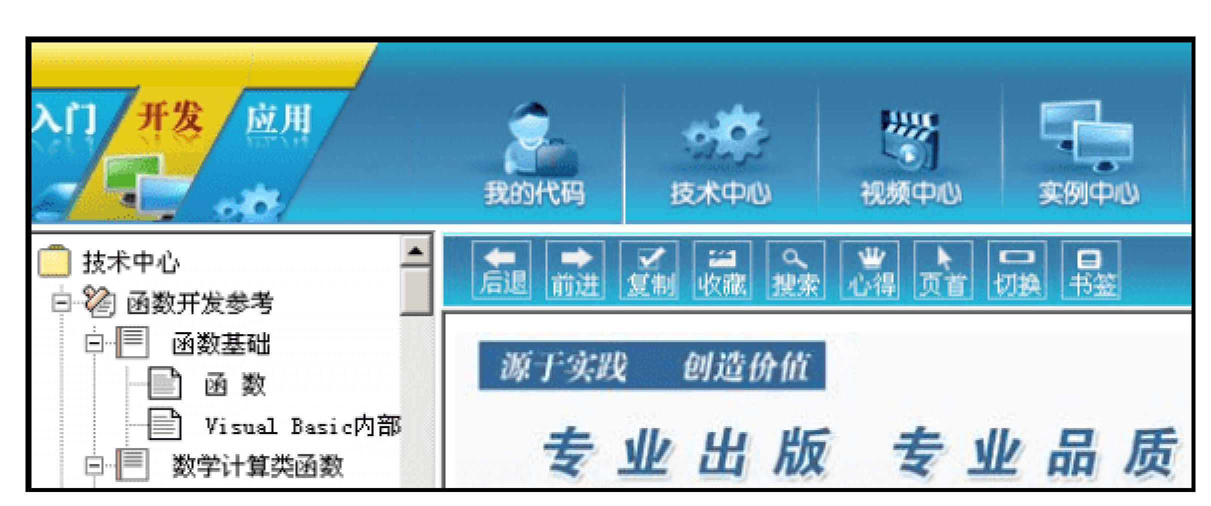


图 12.20　开发类的内容树状导航







如果是应用类，内容树状导航中默认展示的是“源码管理”功能模块的内容，如图 12.21 所示。




图 12.21　应用类的内容树状导航

12.5.2　内容树状导航实现过程

在内容树状导航概述中已经了解，内容树状导航除包含 pd_left.php 和 pd_left.html 文件外，还包含入门、开发和应用 3 类中的 6 个文件。其具体实现过程如下。

（1）在 pd_left.php 文件中，获取网站模块导航的超链接中传递的参数值，并且将参数值赋给 SESSION 变量，然后通过 switch 语句，根据传递的参数值进行判断，进而控制框架中 topwindow（网站模块导航）、leftwindow（内容树状导航）和 mainwindow（内容展示区）3 部分内容的输出。其关键代码如下：

```
<?php
session_start();                                         //初始化 SESSION 变量
header("content-type:text/html; charset=gb2312");
if(!isset($_GET['conn_id']) && !isset($_GET['table_name'])){  //判断 conn_id 和 table_name 的值为空
	$_SESSION['conn_id']="20510111";                 //设置初始值
	$_SESSION['table_name']="study";                 //设置数据库名称
	$_SESSION['class_type']="从零开始";              //设置功能模块
	$conn_id="20510111";
}else{                            //如果不为空，则直接将它们的值赋给 SESSION 变量
	$_SESSION['conn_id']=$_GET['conn_id'];
	$_SESSION['table_name']=$_GET['table_name'];
	$_SESSION['class_type']=$_GET['class_type'];
	$conn_id=$_GET['conn_id'];
}
require 'system/system.inc.php';    //包含数据库连接、操作方法和 Smarty 的配置方法
/*
switch 条件控制语句，判断 conn_id 的值是否与 case 关键字
中指定的值相同，如果相同，则执行相应的操作
*/
switch($conn_id){
	case "20510111":                                 //指定关键字
		include 'accidence.php';                 //包含文件
		$smarty->assign('phtml','accidence.html');   //将模板文件的名称赋给模板变量
		//通过 Script 脚本向指定的文件中传递参数
		echo
```





```
"<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";
            echo            "<script>parent.mainwindow.location.href='pd_main.php?conn_id=
20510111';</script>";
            break;                                  //跳出循环
        case "20610111":
            include 'accidence.php';
            $smarty->assign('phtml','accidence.html');
            echo            "<script>parent.mainwindow.location.href='pd_main.php?conn_id=
26010111';</script>";
            break;
        …//省略了部分代码
        default:
            include 'accidence.php';
            $smarty->assign('phtml','accidence.html');
            echo
"<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";
    echo                "<script>try{parent.mainwindow.location.href='pd_main.php?conn_id=
20510111';}catch(e){}</script>";
            break;
    }
    $smarty->display('pd_left.html');                       //指定模板页
    ?>
```

（2）pd_left.html 模板文件中输出的内容由 switch 语句中定义的模板变量 phtml 的值决定，在 pd_left.html 文件中应用 Smarty 中的 include 函数包含由 phtml 指定的模板文件。其关键代码如下：

```
<script language="javascript" src="js/left.js"></script>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
<div style="width: 100%; height: 100%; padding-left:5px; padding-top:5px;">
{include file=$phtml}
</div>
```

注意：

在 pd_left.html 模板文件中，必须载入 left.js 脚本文件和 link.css 样式文件，它们是实现树状导航菜单功能和控制页面属性的关键。

至于入门、开发和应用 3 类中的 6 个文件，其实现原理是相同的，这里以开发类中的 developer.html 和 developer.php 为例，讲解其实现步骤。

（3）定义 developer.php 文件，获取 SESSION 变量中传递的值，执行查询操作，将查询结果赋给指定的模板变量；注册模板函数，读取指定的文件。其代码如下：

```
<?php
if(!isset($_SESSION['conn_id']) and isset($_SESSION['table_name'])){     //判断 SESSION 变
量的值是否为空
    $conn_id="10127105";        //如果 SESSION 变量的值为空，则直接为变量赋值
    $table_name="ymzx";
```



```
		$class_type="源码管理";
	}else{		//如果 SESSION 变量的值不为空，则将 SESSION 变量的值赋给指定的变量
		$conn_id=$_SESSION['conn_id'];
		$table_name=$_SESSION['table_name'];
		$class_type=$_SESSION['class_type'];
	}
		$smarty->assign("conn_id",$conn_id);				//设置模板变量
		$smarty->assign("class_type",$class_type);			//设置模板变量
		function unHtml($params) {					//创建自定义函数
			extract($params);					//将数组中的数据转换到变量中
			if(strlen($text)==12){				//判断字符串的长度是否等于 12
				$str = is_file("files/".$text.".htm");		//判断指定的文件是否存在
				$str1 = is_file("files/".$text.".exe");		//判断指定的文件是否存在
				if($str){					//如果文件存在
					return "files/".$text.".htm";		//返回文件名
				}else if($str1){
					return "files/".$text.".exe";		//返回文件名
				}else{
					return "files/1999.htm";		//否则，返回一个新的文件
				}
			}else if(strlen($text)==16){			//判断字符串的长度是否等于 16
				$str = is_file("files/".$text.".htm");		//判断指定的文件是否存在
				if($str){
					return "files/".$text.".htm";		//返回文件名
				}else if($str1){
					return "files/".$text.".exe";		//返回文件名
				}else{
					return "files/2000.htm";		//否则，返回一个新的文件
				}
			}
		}
		$smarty->register_function("Util", "unHtml");		//注册模板函数
		if($conn_id=="10121002@10122105@10123105@10125105"){	//判断变量的值
			$array_id=explode("@",$conn_id);		//将 id 值返回到数组中
			$array_table=explode("@",$table_name);		//将数据表名称的值返回到数组中
			for($i=0;$i<count($array_id);$i++){		//循环读取数组中的 id 值
				//执行查询操作
		$rst[$i] = $admindb->ExecSQL("select * from $array_table[$i] where xxbh like 
'$array_id[$i]%'",$conn);
			}
			$result=array_merge($rst[0],$rst[1],$rst[2],$rst[3]);		//合并查询结果
			$smarty->assign('subject',$result);		//将合并的查询结果赋给模板变量
		}else{
			//执行查询操作
			$rst = $admindb->ExecSQL("select * from $table_name where xxbh like 
'$conn_id%'",$conn);
			$smarty->assign('subject',$rst);		//将查询结果赋给模板变量
		}
	?>
```



Note


Note

（4）创建 developer.html 模板文件，通过 section 循环语句，读取模板变量中存储的数据，并且应用 if 语句进行判断，如果指定字符串的长度是 8 字节，则说明是一级导航菜单；如果指定字符串的长度是 12 字节，并且其前 8 个字节与一级导航菜单的 8 个字节相同，那么它是二级导航菜单；如果指定字符串的长度是 16 字节，并且其前 8 个字节与一级导航菜单的 8 个字节相同，其前 12 个字节与二级导航菜单的 12 个字节相同，那么它是三级导航菜单。developer.html 的关键代码如下：

```
<div style="display:block; width:400px">
{section name=id loop=$subject}
        {if $subject[id].xxbh|count_characters=="8"}
        <div style="padding-left:6px; width:400px; height:16px"
onClick="change('div_{$subject[id].xxbh}','icon_div_{$subject[id].xxbh}');">
                <ul>
                        <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/jian-1.gif)" id="icon_div_{$subject[id].xxbh}">
<img src='images/jian-2.gif' width="16" height="16" /></li>
<li style="display:block; width:16px; height:20px; float:left; font-size:0px; text-align:left;"> {if $subject[id].link!=""}
<img alt="" src="images/developer/02.gif" width="16" height="16"> {else}
<img alt="" src="images/accidence/05.gif" width="16" height="16"> {/if} </li>
                <li style="white-space:nowrap; display:block; width:345px; height:20px; padding-top:3px; float:left;
text-align:left; font-size:12px; "> {$subject[id].name}</li>
                </ul>
        </div>
        <div id="div_{$subject[id].xxbh}" style=" display:block; padding-left:2px; width:400px; height:16px" >
        {section name=id12 loop=$subject}
        {if $subject[id12].xxbh|count_characters=="12" && $subject[id12].xxbh|truncate:8:""==$subject[id].xxbh }
                <div style=" display:block; padding-left:4px; width:400px; height:16px"
onClick="change('div_{$subject[id12].xxbh}','icon_div_{$subject[id12].xxbh}');">
                        <ul>
                                <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/line-down.gif)"></li>
                                <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/jian-1.gif)" id="icon_div_{$subject[id12].xxbh}">
<img src='images/jian-1.gif width="16" height="16" /></li>
<li style="display:block; width:16px; height:20px; float:left; text-align:left; font-size:0px;"> {if $subject[id12].link!=""}
<img alt="" src="images/developer/01.gif" width="16" height="16"> {else}
<img alt="" src="images/accidence/06.gif" width="16" height="16"> {/if} </li>
                                <li onMouseDown="this.style.background='#999999';"
onMouseOut="this.style.background='#FFFFFF'" style="white-space:nowrap; display:block; width:329px; height:20px;
padding-top:3px; float:left; text-align:left; font-size:12px; "> 
```



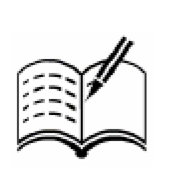
```
            <a href="{Util text=$subject[id12].xxbh}?conn_id={$conn_id}" target="mainwindow">{$subject[id12].name}</a></li>
                </ul>
            </div>
            <div id="div_{$subject[id12].xxbh}" style=" display:block;" >
            {section name=id16 loop=$subject}
                {if $subject[id16].xxbh|count_characters=="16" &&
                $subject[id16].xxbh|truncate:8:""==$subject[id].xxbh && $subject[id16].xxbh|truncate:12:""==$subject[id12].xxbh }
                <div style="padding-left:4px; width:400px; height:16px" >
                    <ul>
                        <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
    background-image:url(images/line-down.gif)"></li>
                        <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
    background-image:url(images/line-down.gif)"></li>
                        <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
    background-image:url(images/line-center-1.gif)"></li>
                <li style="display:block; width:16px; height:20px; float:left; text-align:left; font-size:0px;">
    <img alt="" src="images/accidence/03.gif" width="16" height="16"></li>
                        <li onMouseDown="this.style.background='#999999';"
    onMouseOut="this.style.background='#FFFFFF'" style="white-space:nowrap; display:block; width:313px; height:20px;
    padding-top:3px; float:left; text-align:left; font-size:12px; "> 
            <a href="{Util text=$subject[id16].xxbh}?conn_id={$conn_id}" target="mainwindow">{$subject[id16].name}</a> </li>
                    </ul>
                </div>
                {/if}
            {/section}
            </div>
            {/if}
        {/section}
        </div>
        {/if}
    {/section}
    </div>
```

12.6　内容展示区设计

12.6.1　内容展示区概述

在内容展示区中，输出内容树状导航菜单中指定标题的完整内容。如果指定标题的完整内容不存在，则输出指定的替代文件。其运行效果如图 12.22 所示。





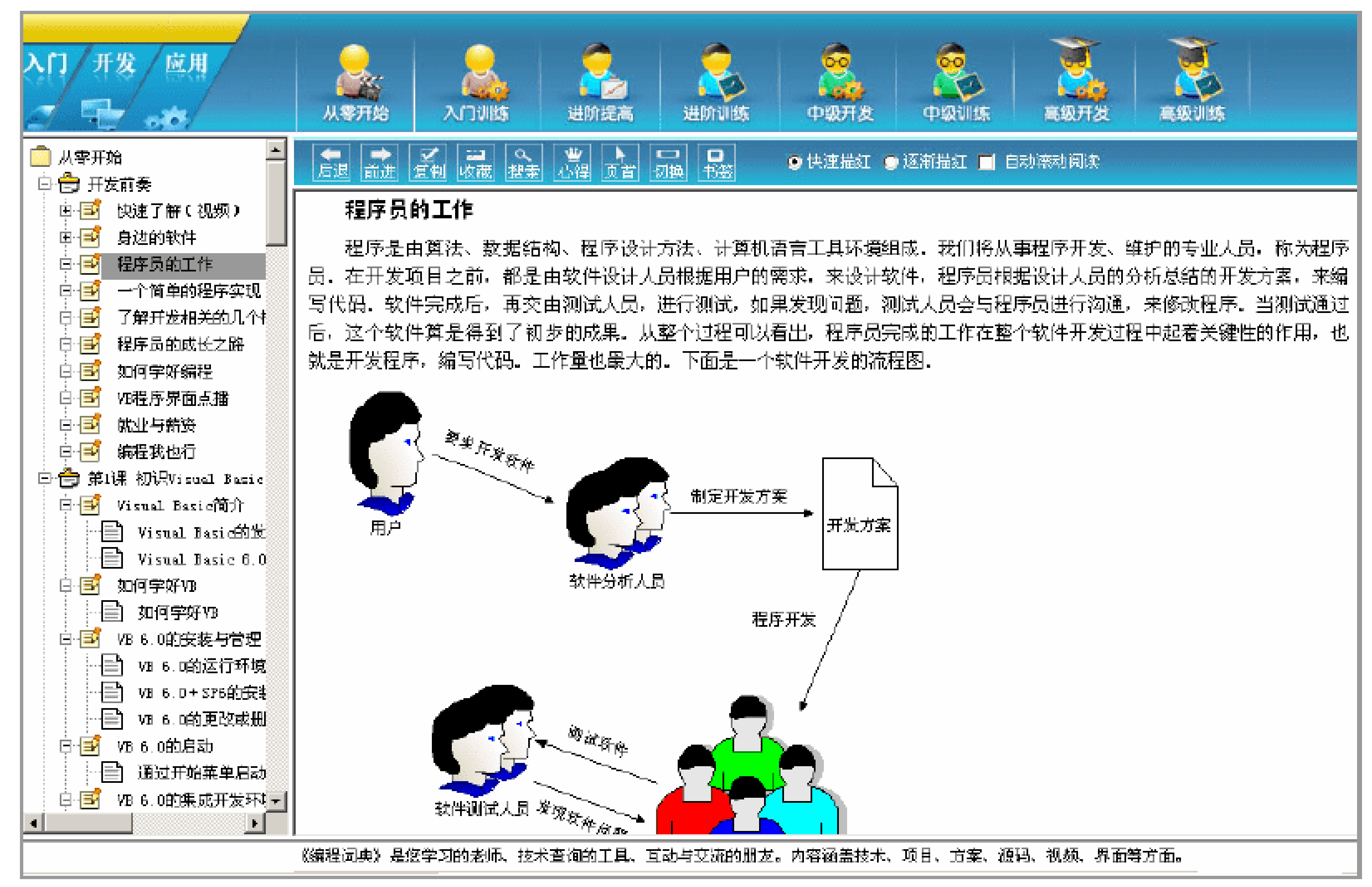


图 12.22　内容展示区

12.6.2　内容展示区实现过程

内容展示区只包含一个文件 pd_main.php。在该文件中判断超链接传递的参数值，并根据该值包含不同的替代文件。其代码如下：

```
<?php
if(isset($_GET['conn_id']) && $_GET['conn_id']=="20510111" ){    //判断超链接的 id 值
        include_once("files/19913.htm");                         //调用指定的包含文件
}else if(isset($_GET['conn_id']) && $_GET['conn_id']=="10121605"){
        include_once("files/19913.htm");                         //调用指定的包含文件
}else{
        include_once("files/2000.htm");                          //调用指定的包含文件
}
?>
```

说明：

至此，有关在线编程词典网站的内容讲解完毕。这里没有对功能导航条和滚动导航条的内容进行详细讲解，因为在这两项中没有具体的内容。

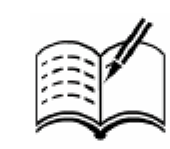
12.7　技 术 提 炼

在在线编程词典模块的开发过程中，多处使用了较新的技术。为了便于对该项目各个模块的理解，下面首先对在线编程词典网站中所使用的关键技术进行详细讲解。

12.7.1　Smarty 模板

在线编程词典网站采用 Smarty 模板实现网站的动静分离。Smarty 的版本是





Smarty-2.6.23，下载后将 libs 文件夹存储于 mr\12\system\文件夹下。

Smarty 的配置方法存储于本项目的 system 文件夹下，名称为 system.smarty.inc.php。配置文件的代码如下：

```
<?php
require("libs/Smarty.class.php");                          //包含模板文件
class SmartyProject extends   Smarty{                      //定义类，继承模板类
    function SmartyProject(){                              //定义方法
        $this->template_dir = "./system/templates/";       //指定模板文件存储位置
        $this->compile_dir = "./system/templates_c/";      //指定编译文件存储位置
        $this->config_dir = "./system/configs/";           //指定配置文件存储位置
        $this->cache_dir = "./system/cache/";              //指定缓存文件存储位置
    }
}
?>
```

提示：

在包含 Smarty 类文件时，必须设置正确的包含路径，否则无法调用 Smarty 类文件（在程序运行过程中，将出现如图 12.23 所示的错误信息）。



图 12.23　Smarty 模板的路径设置不正确

类的实例化操作在 system 文件夹下的 system.inc.php 文件中实现，返回对象名称为 $smarty。在 system.inc.php 文件中还包含了其他类的实例化操作，其代码如下：

```
<?php
require("system.smarty.inc.php");                    //包含 Smarty 配置类
require("system.class.inc.php");                     //包含数据库连接和操作类
$connobj=new ConnDB("mssql","localhost","sa","","db_programme");//数据库连接类实例化
$conn=$connobj->GetConnId();                         //执行连接操作，返回连接标识
$admindb=new AdminDB();                              //数据库操作类实例化
$smarty=new SmartyProject();                         //调用 Smarty 模板
?>
```

至此，Smarty 模板的安装和配置讲解完毕。

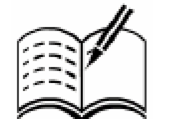
12.7.2　PDO 数据库抽象层

在本项目中，通过 PDO 数据库抽象层连接和操作 SQL Server 数据库。PDO 是与 PHP 5.1 一起发行的，默认包含在 PHP 5.1 中。由于 PDO 需要 PHP 5 核心面向对象特性的支持，因此无法在 PHP 5.0 之前的版本中使用它。



默认情况下，PDO 在 PHP 5.2 中处于开启状态，但是要启用对某个数据库驱动程序的支持，仍需要进行相应的配置操作。

在 Linux 环境下，要使用 MySQL 数据库，可以在 configure 命令中添加如下选项。



Note

```
--with-pdo-mysql=/path/to/mysql/installation
```

在 Windows 环境下，PDO 在 php.ini 文件中进行配置，如图 12.24 所示。



图 12.24　在 Windows 环境下配置 PDO

要启用 PDO，首先必须加载“extension=php_pdo.dll”；要想让其支持某个具体的数据库，则还要加载对应的数据库选项。例如，要支持 MS SQL Server 数据库，则需要加载“extension=php_pdo_mssql.dll”选项。

注意：

在完成数据库的加载后，要保存 php.ini 文件，并且重新启动 Apache 服务器，修改才能够生效。

在本项目中，将 PDO 连接和操作 SQL Server 数据库的方法都封装到类中，通过对类中方法的调用即可完成数据库的连接和管理操作。定义的类文件存储于本项目的 MR\12\system 文件夹下，名称为 system.class.inc.php。

在 ConnDB 类中，定义连接数据库和关闭数据库的方法。其关键代码如下：

```
class ConnDB                                        //数据库连接类
var $dbtype;
var $host;
    var $user;
    var $pwd;
    var $dbname;
    function ConnDB($dbtype,$host,$user,$pwd,$dbname){        //构造方法
     $this->dbtype=$dbtype;
     $this->host=$host;
     $this->user=$user;
     $this->pwd=$pwd;
     $this->dbname=$dbname;
}
function GetConnId(){                               //实现数据库的连接并返回连接对象
     if($this->dbtype=="mysql" || $this->dbtype=="mssql"){
          $dsn="$this->dbtype:host=$this->host;dbname=$this->dbname";
          }else{
               $dsn="$this->dbtype:dbname=$this->dbname";
```






```
            }
            try {
        $conn = new PDO($dsn, $this->user, $this->pwd); //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
                $conn->query("set names gb2312");
            return $conn;
            } catch (PDOException $e) {
            die ("Error!: " . $e->getMessage() . "<br/>");
            }
        }
    }
```

在 AdminDB 类中，定义数据库的操作方法。其关键代码如下：

```
class AdminDB{                                      //数据库管理类
    function ExecSQL($sqlstr,$conn){
        $sqltype=strtolower(substr(trim($sqlstr),0,6));
        $rs=$conn->prepare($sqlstr);                //准备查询语句
        $rs->execute();                             //执行查询语句，并返回结果集
        if($sqltype=="select"){
            $array=$rs->fetchAll(PDO::FETCH_ASSOC);//获取结果集中的所有数据
            if(count($array)==0 || $rs==false)
                return false;
            else
                return $array;
        }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
            if($rs)
                return true;
            else
                return false;
        }
    }
}
```

类的实例化文件，都存储于 system 文件夹下的 system.inc.php 文件中。数据库连接成功返回的标识是$conn，而数据库操作类实例化的对象是$admindb。

12.7.3　树状导航菜单设计

为了方便浏览者查看在线编程词典网站的海量数据资源，这里通过树状导航菜单输出资源列表，其运行效果如图 12.25 所示。

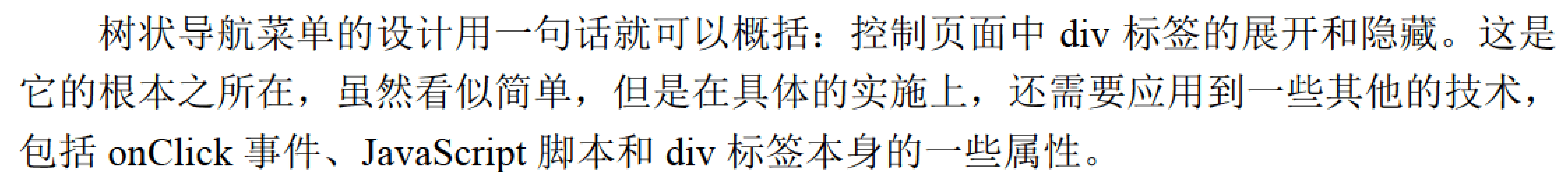


图 12.25　通过树状导航菜单输出数据资源信息



那么，这个树状导航菜单是如何实现的呢？其开发的原理又是怎么样的呢？下面就来对这些疑问一一进行解答。

1．设计原理



树状导航菜单的设计用一句话就可以概括：控制页面中 div 标签的展开和隐藏。这是它的根本之所在，虽然看似简单，但是在具体的实施上，还需要应用到一些其他的技术，包括 onClick 事件、JavaScript 脚本和 div 标签本身的一些属性。

2．具体方案

树状导航菜单的具体开发步骤如下：

（1）创建 div 标签，根据从模板变量中读取的数据，设置 div 标签的 id，设置 div 标签样式 style 中 display 的值为 block。

（2）在 div 标签中通过 onClick 事件调用 JavaScript 脚本中的方法，并将定义的 div 标签的 id 值作为参数。

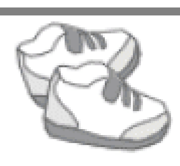
上述两步操作都是在 Smarty 的模板文件中完成的。下面以 accidence.html 模板文件为例，给出其具体代码。

```
<div style="display:block; width:400px">
{section name=id loop=$subject}
	{if $subject[id].xxbh|count_characters=="12"}
❶	<div style="padding-left:6px; width:400px; height:16px"
onClick="change('div_{$subject[id].xxbh}','icon_div_{$subject[id].xxbh}');">
		<ul>
❷			<li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/jian-1.gif)" id="icon_div_{$subject[id].xxbh}">
<img src='images/jian-2.gif' width="16" height="16" /></li>
<li style="display:block; width:16px; height:20px; float:left; font-size:0px; text-align:left;"> {if $subject[id].link!=""}
<img alt="" src="images/accidence/01.gif" width="16" height="16"> {else}
<img alt="" src="images/accidence/05.gif" width="16" height="16"> {/if} </li>
		<li style="white-space:nowrap; display:block; width:345px; height:20px; padding-top:3px; float:left;
text-align:left; font-size:12px; "> {$subject[id].name}</li>
		</ul>
	</div>
❸	<div id="div_{$subject[id].xxbh}" style=" display:block; padding-left:2px; width:400px; height:16px" >
	{section name=id16 loop=$subject}
	{if $subject[id16].xxbh|count_characters=="16" && $subject[id16].xxbh|truncate:12:""==$subject[id].xxbh }
❹		<div style=" display:block; padding-left:4px; width:400px; height:16px"
onClick="change('div_{$subject[id16].xxbh}','icon_div_{$subject[id16].xxbh}');">
			<ul>
				<li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/line-down.gif)"></li>
```






```
❺           <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/jian-1.gif)" id="icon_div_{$subject[id16].xxbh}">
<img src='images/jian-1.gif' width="16" height="16" /></li>
            <li style="display:block; width:16px; height:20px; float:left; text-align:left; font-size:0px;">
{if $subject[id16].link!=""} <img alt="" src="images/accidence/02.gif" width="16" height="16"> {else}
 <img alt="" src="images/accidence/06.gif" width="16" height="16"> {/if} </li>
<li onMouseDown="this.style.background='#999999';" onMouseOut="this.style.background='#FFFFFF'"
style="white-space:nowrap; display:block; width:329px; height:20px; padding-top:3px; float:left; text-align:left;
font-size:12px; ">  <a href="{Util text=$subject[id16].xxbh}?conn_id={$conn_id}"
target="mainwindow">{$subject[id16].name}</a></li>
                </ul>
            </div>
❻           <div id="div_{$subject[id16].xxbh}" style=" display:block;" >
            {section name=id20 loop=$subject}
                {if $subject[id20].xxbh|count_characters=="20" &&
               $subject[id16].xxbh|truncate:12:""==$subject[id].xxbh && $subject[id20].xxbh|truncate:16:""==$subject[id16].xxbh }
                <div style="padding-left:4px; width:400px; height:16px" >
                    <ul>
                        <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/line-down.gif)"></li>
                        <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/line-down.gif)"></li>
                        <li style="display:block; width:16px; height:20px; float:left; font-size:0px;
background-image:url(images/line-center-1.gif)"></li>
                <li style="display:block; width:16px; height:20px; float:left; text-align:left; font-size:0px;">
<img alt="" src="images/accidence/03.gif" width="16" height="16"></li>
                        <li onMouseDown="this.style.background='#999999';"
onMouseOut="this.style.background='#FFFFFF'" style="white-space:nowrap; display:block; width:313px; height:20px;
padding-top:3px; float:left; text-align:left; font-size:12px; ">  <a href="{Util
text=$subject[id20].xxbh}?conn_id={$conn_id}" target="mainwindow">{$subject[id20].name}</a> </li>
                    </ul>
                </div>
                {/if}
            {/section}
            </div>
            {/if}
        {/section}
```



```
        </div>
        {/if}
    {/section}
    </div>
```



Note

关键代码解析

❶ onClick 事件：调用 change 方法，将在❷和❸中定义的 id 值作为参数传递。

❷ id="icon_div_{$subject[id].xxbh}"：设置 li 标签的 id 值，并设置其样式 style 中的 display 的值为 block。

❸ id="div_{$subject[id].xxbh}"：设置 div 标签的 id 值，并设置其样式 style 中的 display 的值为 block。

❹ onClick 事件：调用 change 方法，将在❺和❻中定义的 id 值作为参数传递。

❺ id="icon_div_{$subject[id16].xxbh}"：设置 li 标签的 id 值，并设置其样式 style 中的 display 的值为 block。

❻ id="div_{$subject[id16].xxbh}"：设置 div 标签的 id 值，并设置其样式 style 中的 display 的值为 block。

说明：

在本项目的入门、开发和应用 3 个模块中都用到了树状导航菜单功能，这里仅以入门模块为例对其进行讲解。其中涉及到的 Smarty 模板技术已在模块中进行详细讲解。这里之所以应用了两次 onClick 事件，是因为实现的是一个 3 级导航的菜单。

（3）创建 JavaScript 脚本文件 left.js，定义 change 方法，根据传递的参数值控制 div 标签样式 style 中 display 的值。change 方法的代码如下：

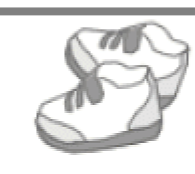
```
function change(nu,icon){                    //定义 change 方法，包含两个参数
    nu=document.getElementById(nu);          //获取参数 nu 的 ID 值
    icon=document.getElementById(icon);      //获取参数 icon 的参数值
    if(nu.style.display == "none"){          //判断，如果参数 nu 的样式为 none（隐藏）
        nu.style.display = "block";          //则设置 nu 的样式为 block（展开）
        icon.innerHTML = "<img src='images/jian-1.gif' width='16' height='16' />";      //输出图片
    }else{
        nu.style.display = "none";           //如果 nu 的样式为隐藏，
        icon.innerHTML = "<img src='images/jia-1.gif' width='16' height='16' />";//输出图片
    }
}
```

12.7.4 向 Flash 中传递参数

在本程序的 LOG 中，应用 Flash 实现入门、开发和应用 3 个模块之间的跳转，并通过 Flash 完成参数的传递，其运行效果如图 12.26 所示。



图 12.26　通过 Flash 完成模块之间的跳转





下面一起探讨一下如何通过 Flash 传递参数。

（1）创建 JS 脚本文件 flash1.js，定义向 Flash 中写入的内容。其代码如下：

```
function flash1(m){
    var flash2='<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=
7,0,19,0"
width="171" height="93">';
❶          flash2+='<param    id="swf"    name="movie"    value="images/bccd.swf?conn_type
='+m+'">';
        flash2+='<param name="quality" value="high">';
        flash2+='<param name="allScriptAccess" value="always"/>';
        flash2+='<param name="wmode" value="transparent">';
        flash2+='<embed src="images/bccd.swf" width="171" height="93" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash"
wmode="transparent"></embed>';
        flash2+='</object>';
❷     document.write(flash2);
}
```

关键代码解析

❶ param id="swf"：在 swf 的 url 中传递参数值。

❷ document.write：执行写入操作。

（2）在 Flash 元件中通过 on(release)方法获取传递的参数值，并执行超链接。

① 通过 this.loaderinfo.parameters["conn_type"]获取传递的参数值；

```
var type=this.loaderinfo.parameters["conn_type"];
```

在 Flash 编辑器中的编辑效果如图 12.27 所示。

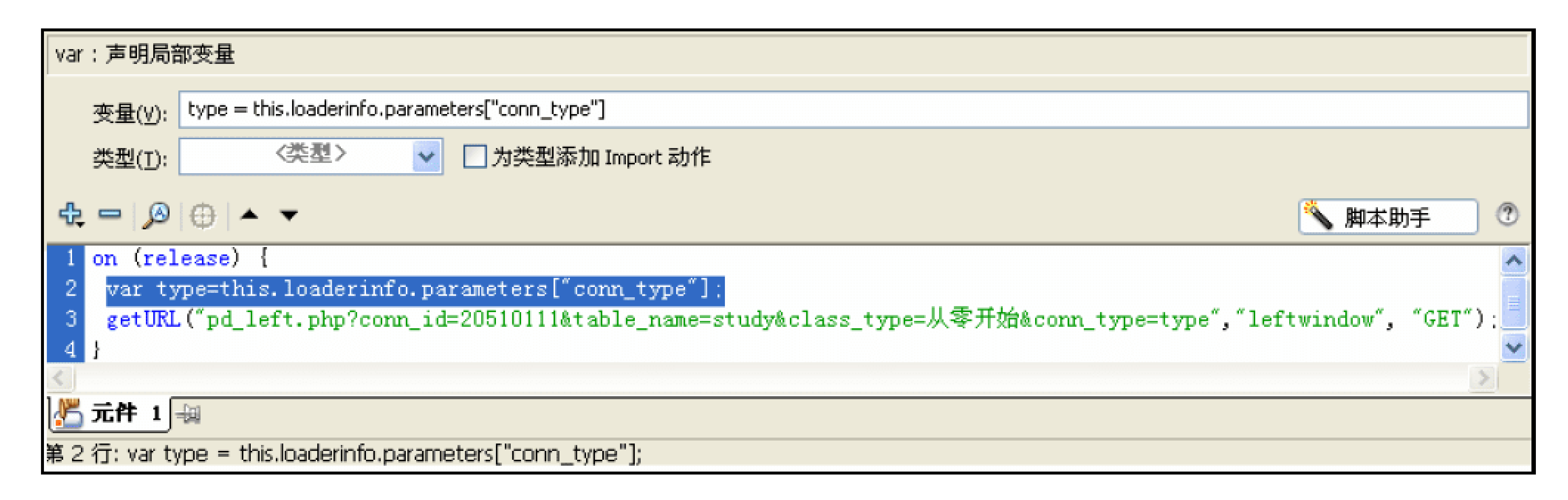
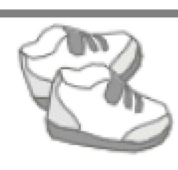


图 12.27　通过变量传递参数

② 通过 getURL()方法执行超链接。

```
getURL("pd_left.php?conn_id=20510111&table_name=study&class_type= 从 零 开 始 &conn_type=
type", "leftwindow", "GET");
```

在 Flash 编辑器中的编辑效果如图 12.28 所示。



Note

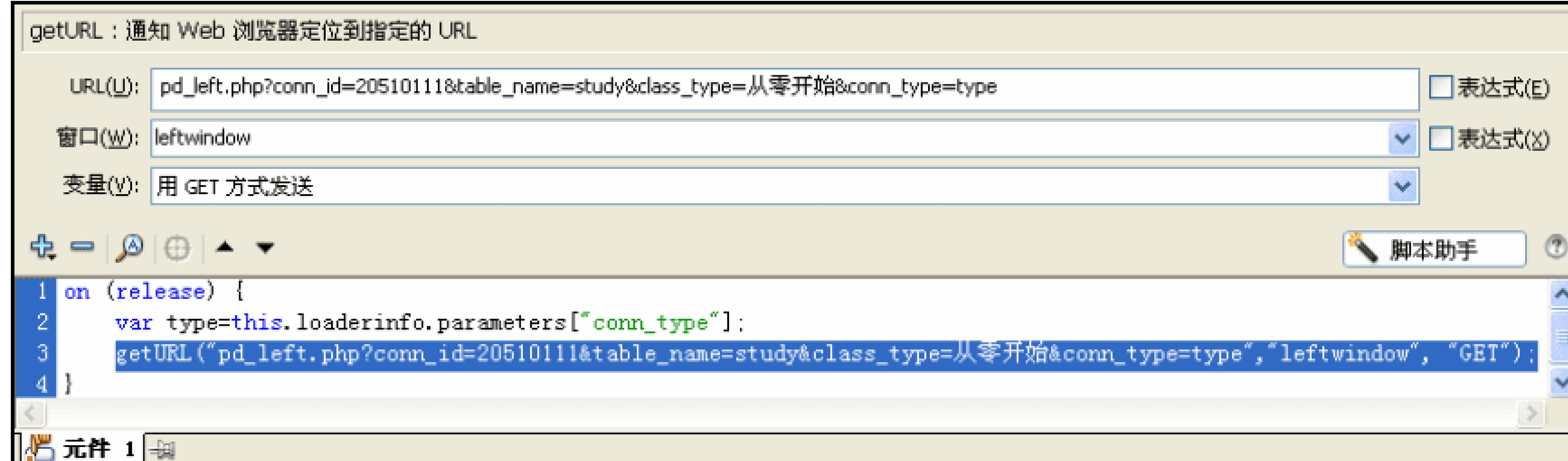


图 12.28　通过 getURL 执行超链接

（3）在本程序使用的 FLASH 中，创建了 3 个元件，分别对应入门、开发和应用 3 个模块。

元件 1 的脚本代码如下：

```
on (release) {
    var type=this.loaderinfo.parameters["conn_type"];
    getURL("pd_left.php?conn_id=20510111&table_name=study&class_type=从零开始&conn_type=type",
"leftwindow", "GET");
}
```

元件 2 的脚本代码如下：

```
on (release) {
    var type=this.loaderinfo.parameters["conn_type"];
    getURL("pd_left.php?conn_id=10121002&table_name=jszx&class_type=技术中心&conn_type=type",
"leftwindow", "GET");
}
```

元件 3 的脚本代码如下：

```
on (release) {
    var type=this.loaderinfo.parameters["conn_type"];
    getURL("pd_left.php?conn_id=10127105&table_name=ymzx&class_type=源码管理&conn_type=type",
"leftwindow", "GET");
}
```

12.7.5　Smarty 注册模板函数——读取文件

在本程序中，不仅使用 Smarty 模板实现网站的动静分离，而且应用了 Smarty 的注册模板函数，判断指定的文件是否存在，如果存在则返回文件名，否则返回一个替代文件名。

注册模板函数存储于 accidence.php 中，其代码如下。

```
function unHtml($params) {                          //创建自定义函数
    extract($params);                               //将数组中的数据转换到变量中
    if(strlen($text)==16){                          //判断当参数值的长度为 16 字节时
        $str = is_file("files/".$text.".htm");      //判断指定文件是否存在
```



```
                if($str){
                    return "files/".$text.".htm";       //返回指定的文件
                }else{
                    return "files/19913.htm";           //如果指定文件不存在，则输出该文件
                }
            }else if(strlen($text)==20){                //判断当参数值的长度为 20 字节时
                $str = is_file("files/".$text.".htm");  //判断指定文件是否存在
                if($str){
                    return "files/".$text.".htm";       //返回指定的文件
                }else{
                    return "files/2000.htm";            //如果指定文件不存在，则输出该文件
                }
            }
        }
        $smarty->register_function("Util", "unHtml");   //注册模板函数
```

在模板页中，调用注册模板函数的方法如下。

```
    <a href="{Util text=$subject[id20].xxbh}?conn_id={$conn_id}" target="mainwindow">
{$subject[id20].name}</a>
```

12.7.6 FRAME 框架技术

在线编程词典网站首页应用框架设计，共分为 5 个部分，如图 12.29 所示。

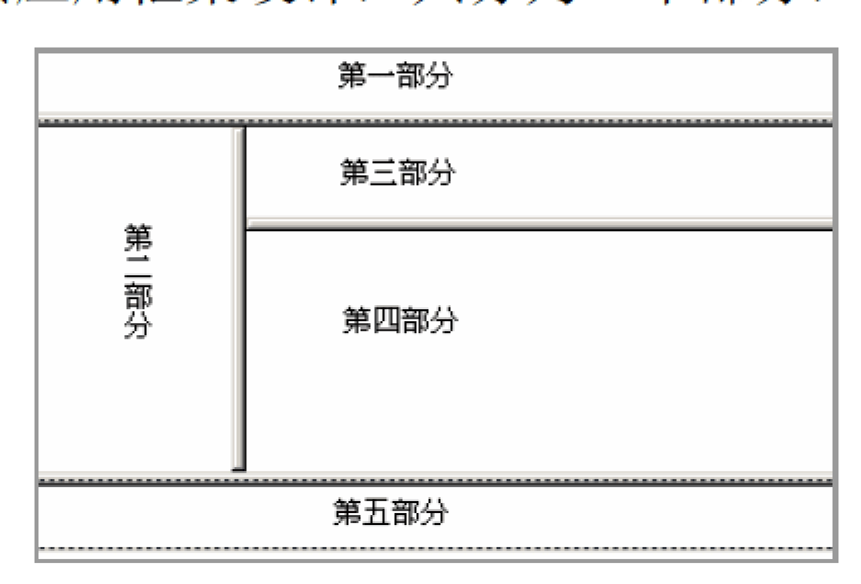
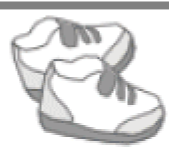


图 12.29 在线编程词典网站布局

index.html 框架页的代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>{$conn_type}编程词典</title>
</head>
<link rel="stylesheet" type="text/css" href="css/link.css">
<frameset rows="93,*,25" frameborder="0">
<frame src="pd_top.php?conn_type={$conn_type}" name="topwindow" scrolling="no">
    <frameset cols="200,*">
        <frame src="pd_left.php?conn_type={$conn_type}" name="leftwindow"
frameborder="1">
        <frameset rows="37,*" frameborder="0">
          <frame src="pd_title.php" name="titlewindow" scrolling="no">
```




```
                <frame src="pd_main.php" name="mainwindow" scrolling="auto" frameborder="1">
            </frameset>
        </frameset>
    <frame src="pd_bottom.php" name="bottomwindow" scrolling="no" frameborder="1">
</frameset><noframes></noframes>
```



Note

框架中的文件分配如下。

☑ 框架主体文件：index.html、index.php。

☑ 网站模块导航：pd_top.html、pd_top.php。

☑ 内容树状导航：pd_left.html、pd_left.php。

☑ 功能导航条：pd_title.html、pd_title.php。

☑ 内容展示区：pd_main.html、pd_main.php。

☑ 滚动导航条：pd_bottom.html、pd_bottom.php。

12.7.7 超链接中的图片切换技术

网站模块导航中包含两项技术：

☑ 通过 JS 脚本向 Flash 中传递参数。该技术已经在前文作了详细的讲解，这里不再赘述。

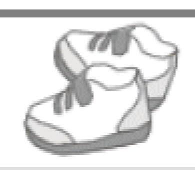
☑ 在超链接中应用 onMouseOver 和 onMouseOut 事件调用 JavaScript 脚本中的方法，控制鼠标滑过超链接图片时的样式。

超链接的设置如下：

```
<a href="pd_left.php?conn_id=20510111&table_name=study&class_type=从零开始" target="leftwindow">
<img src="images/accidence/images_01.jpg" border="0" id="Image1"
onMouseOver="MM_swapImage('Image1','','images/accidence/images_001.jpg',1)"
onMouseOut="MM_swapImgRestore()" /></a>
```

JavaScript 脚本文件 top.js 中定义的方法如下：

```
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}
```






```
function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
```

12.7.8 Smarty 模板技术

在内容树状导航中，系统根据网站模块导航中传递的参数值，控制着框架中 topwindow（网站模块导航）、leftwindow（内容树状导航）和 mainwindow（内容展示区）3 部分内容的输出。其应用到的技术如下：

1. switch 条件控制语句的充分应用

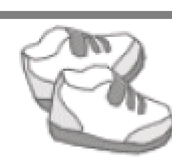
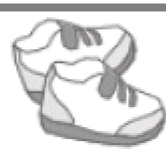
在 pd_left.php 文件中，通过$_GET[]方法获取网站模块导航中参数 conn_id 传递的值，将该值作为 switch 语句执行的条件，与 case 关键字指定的值进行比较，如果相同则执行相应的操作。首先，将一个模板页赋给一个模板变量；然后，通过 script 标签向 topwindow 中的 pd_top.php（网站模块导航）文件中传递参数；最后，通过 script 标签向 mainwindow 中的 pd_main.php（内容展示区）文件中传递参数。其关键代码如下：

```
<?php
/*
switch 条件控制语句，判断 conn_id 的值是否与 case 关键字
中指定的值相同，如果相同，则执行相应的操作
*/
    switch($conn_id){
    case "20510111":                                    //指定关键字
        include 'accidence.php';                        //包含文件
        $smarty->assign('phtml','accidence.html');      //将模板文件的名称赋给模板变量
        //通过 script 脚本向指定的文件中传递参数
        echo "<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";
        echo "<script>parent.mainwindow.location.href='pd_main.php?conn_id=20510111';</script>";
        break;                                          //跳出循环
…//省略了部分代码
```

2. 通过 Smarty 中的 include()函数在当前模板中包含其他模板文件

在 switch 语句中，将指定的模板文件名赋给一个模板变量 phtml。此时在内容树状导航的模板文件 pd_left.html 中，获取模板变量 phtml 传递的值，并且将该值作为 Smarty 中 include()函数包含的模板文件。内容树状导航模板文件 pd_left.html 的关键代码如下：

```
<script language="javascript" src="js/left.js"></script>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
```

```
<body>
<div style="width: 100%; height: 100%; padding-left:5px; padding-top:5px;">
{include file=$phtml}
</div>
</body>
```



3. 通过 Smarty 中的注册模板函数读取指定文件

这里应用 12.7.5 节中讲解的 Smarty 注册模板函数读取指定的文件。详细内容参见 12.7.5 节，这里不再赘述。

4. 通过 Smarty 中的 count_characters 和 truncate 方法统计和截取字符串

在入门类、开发类和应用类的树状导航菜单中，都应用到 Smarty 中的 count_characters 和 truncate 方法，对指定字符串的长度进行统计和截取，进而判断数据属于几级菜单。count_characters 和 truncate 方法在本项目中应用的关键代码如下：

```
{if $subject[id12].xxbh|count_characters=="12" && $subject[id12].xxbh|truncate:8:""==$subject [id].xxbh }
```

上述代码取自 developer.html 模板文件。

5. 通过树状导航菜单展示功能模块的内容

有关树状导航菜单的内容参见 12.5.1 节，这里不再赘述。

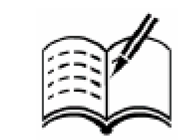
12.7.9　文件检测技术

内容展示区中完整内容的输出，关键是内容树状导航菜单中标题的超链接值的设置。这个超链接的值是从注册模板函数中获取的。也就是说在注册模板函数中判断指定标题的完整文件是否存在，如果存在则输出这个文件，否则输出指定的替代文件。在本项目中定义了 3 个注册模板函数，其功能都是输出指定的文件，这 3 个文件分别存储于 accidence.php、criterion.php 和 developer.php 中，下面给出在 developer.php 文件中具体的文件检测代码。

```
function unHtml($params) {                          //创建自定义函数
    extract($params);                               //将数组中的数据转换到变量中
    if(strlen($text)==12){                          //判断字符串的长度是否等于 12
        $str = is_file("files/".$text.".htm");      //判断指定的文件是否存在
        $str1 = is_file("files/".$text.".exe");     //判断指定的文件是否存在
        if($str){                                   //如果文件存在
            return "files/".$text.".htm";           //返回文件名
        }else if($str1){
            return "files/".$text.".exe";           //返回文件名
        }else{
            return "files/1999.htm";                //否则，返回一个新的文件
        }
    }else if(strlen($text)==16){                    //判断字符串的长度是否等于 16
        $str = is_file("files/".$text.".htm");      //判断指定的文件是否存在
        if($str){
            return "files/".$text.".htm";           //返回文件名
```



```
                }else if($str1){
                    return "files/".$text.".exe";        //返回文件名
                }else{
                    return "files/2000.htm";             //否则，返回一个新的文件
                }
            }
        }
        $smarty->register_function("Util", "unHtml");    //注册模板函数
```



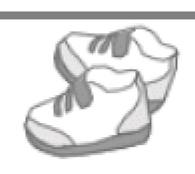
第13章

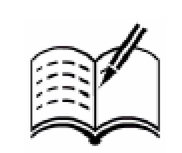
邮件发送系统模块

（自学视频、源程序：配套资源\mr\13\）

电子邮件是 Internet 中应用最广泛的服务之一。通过电子邮件系统，用户可以以极低廉的价格轻松、快捷地与世界上任何一个角落的网络用户进行沟通、交流。电子邮件的内容多种多样，可以是文字、图像、声音等。在构建企业站点时，可以为企业用户提供 E-mail 通信功能，从而极大地提高企业的工作效率，减轻用户的工作负担。电子邮件的收发技术是 PHP 程序员必须掌握的技术之一。通过本章的学习，读者能够学到以下内容。

- 利用 mail()函数发送电子邮件
- 利用 fsockopen()函数接收电子邮件
- 带附件邮件的发送与下载





13.1　邮件发送系统概述

随着 Internet 技术的飞速发展，越来越多的人选择网络来进行即时沟通。本节将要讲解的邮件发送系统涵盖了从登录邮件服务器，到发送邮件、接收邮件、查看邮件内容以及删除邮件等一系列功能的实现方法。另外，还详细介绍了 WinWebMail 的安装和配置方法。

13.1.1　电子邮件简介

电子邮件（E-mail）是 Internet 中应用最广泛的服务之一，具有快速、便捷、低成本等优势，特别适合网络上的信息交流。有了电子邮件系统，用户便可以通过从因特网服务商（ISP）那里申请的账号或邮箱进行信件的收发、访问等操作，轻松、快捷地实现与世界各地的朋友进行沟通、交流。

电子邮件地址的格式是 USER@SERVER.COM。其中包括 3 部分：第 1 部分 USER 代表用户邮箱的账号，对于同一个邮件接收服务器来说，这个账号必须是唯一的；第 2 部分@是分隔符；第 3 部分 SERVER.COM 是用户邮箱的邮件接收服务器域名，用以标识其所在的位置。

13.1.2　功能结构

本章将要开发的邮件发送系统的所有信息都保存在 POP3 服务器上，所以不涉及后台管理及后台数据库。该系统的功能结构如图 13.1 所示。

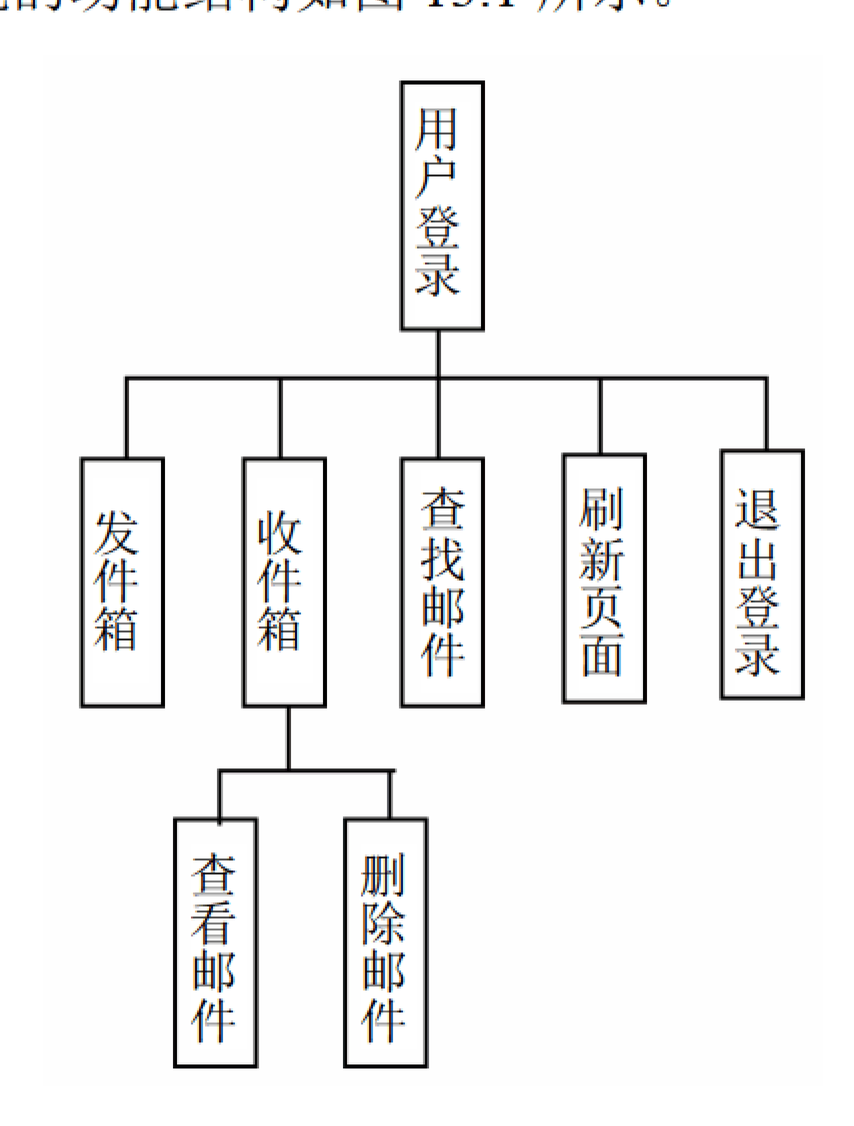
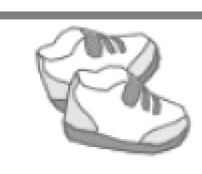
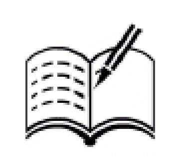


图 13.1　邮件发送系统功能结构图

13.1.3　程序预览

为了让读者对本系统有个初步的了解和认识，下面给出本系统的几个页面运行效果





图，如果想查看完整的效果图，参见配套资源源程序。

登录邮件服务器的页面如图 13.2 所示。



图 13.2　登录邮件服务器

邮件发送系统中发送带附件邮件的页面运行效果如图 13.3 所示。



图 13.3　发送带附件的邮件

邮件发送系统中收件箱的运行效果如图 13.4 所示。



图 13.4　收件箱的运行效果

邮件系统中查看邮件详细信息的页面运行效果如图 13.5 所示。





图 13.5　查看邮件的详细信息



13.2　邮件收发的原理

在具体讲解邮件发送系统功能之前，先来了解一下邮件发送系统发送邮件、接收邮件、查找邮件及删除邮件实现的基本原理。

发送邮件实现相对简单，用 PHP 自身提供的 mail()函数就可以完成该功能。需要注意的是，由于 mail()函数自身不能进行 SMTP 认证，因此需要安装一个 SMTP 服务器转发邮件，并且需要在 php.ini 中设置 SMTP 服务器地址及端口号。

接收邮件需要利用 fsockopen()函数来实现。首先利用该函数连接 POP3 服务器并返回一个 Socket 句柄，之后就可以利用 fputs()函数将 POP 命令发送给服务器，利用 fgets()函数接收 POP 服务器反馈回来的信息（如邮件头、邮件内容），从邮件头中就可以提取发件人地址、发件时间等信息。

查找邮件时，系统首先遍历所有 POP 服务器中的信件，并提取所有信件的主题和发件人地址，如果发现与用户查找的主题或发件人地址相同的邮件，就将该邮件显示出来。

删除邮件只需用 fputs()函数将 POP 命令 DELE 写入服务器即可。需要注意的是，删除邮件必须由命令 QUIT 来执行，否则即使发送了 DELE 命令也无法完成删除工作。

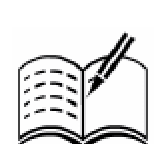
以上内容为整个邮件发送系统实现的核心思想。在此笔者将收发邮件的各项功能封装到 POP3 类中，这样既符合 PHP 面向对象的发展趋势，又为系统带来了很好的移植性，并且为日后的维护工作带来很大方便。

13.3　通过 WinWebMail 配置邮件服务器

在本模块中，我们通过第三方的 WinWebMail 邮件服务器，在本地测试电子邮件系统的功能。下面介绍一下 WinWebMail 的安装和配置。

（1）打开如图 13.6 所示的“WinWebMail 3.7.7.1 标准版”安装界面，单击 Next 按钮。





Note

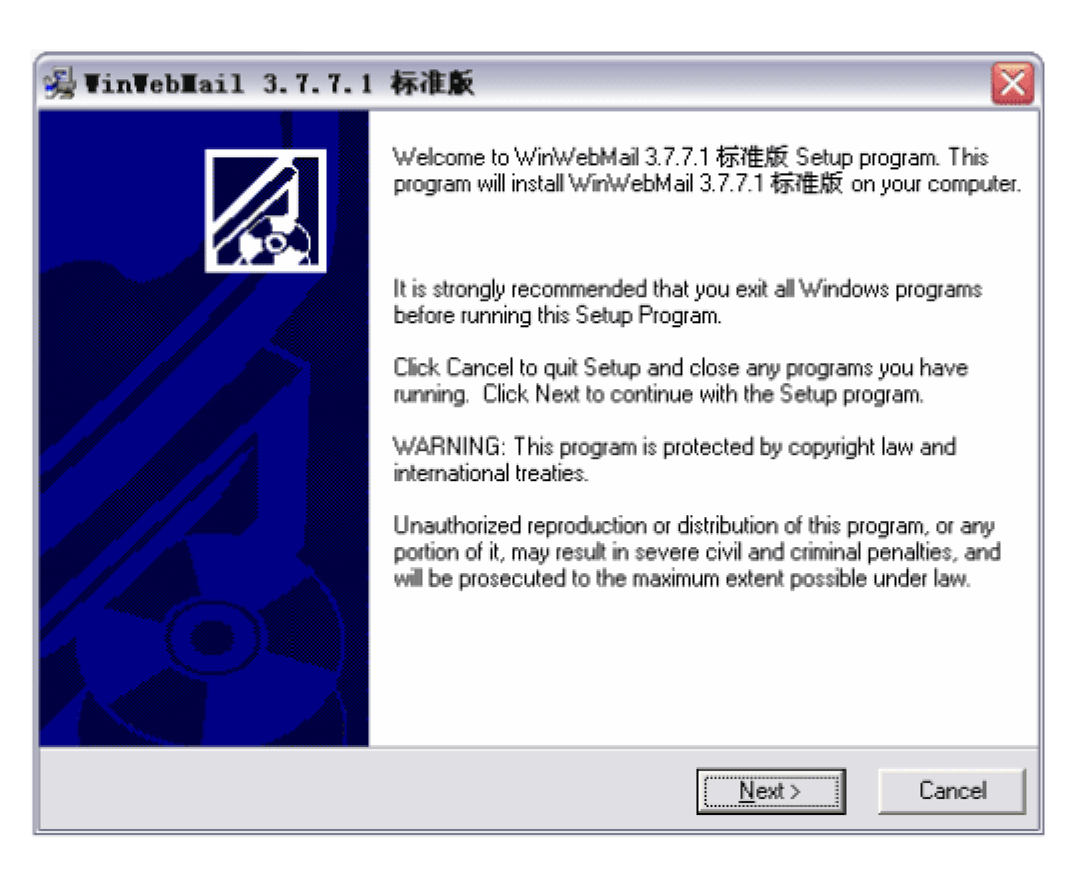


图 13.6 “WinWebMail 3.7.7.1 标准版”安装界面

（2）弹出如图 13.7 所示“WinWebMail 软件产品最终用户许可协议”对话框，单击 Next 按钮。

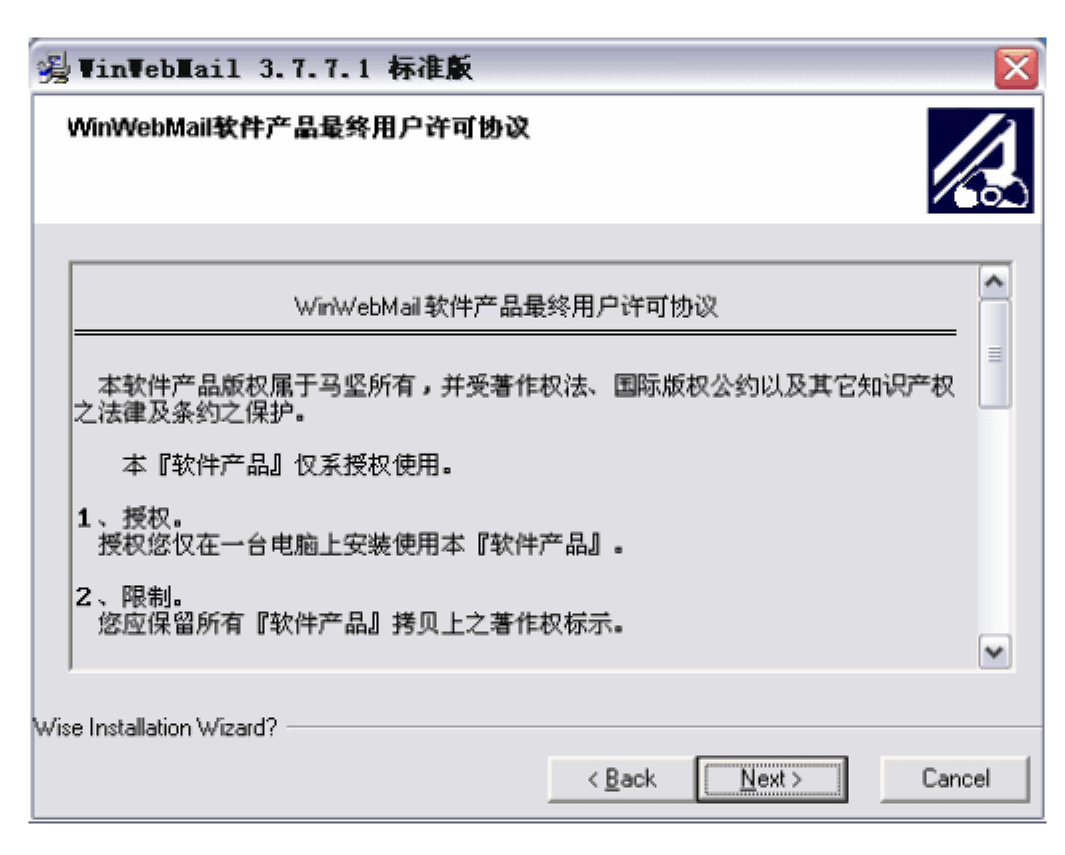


图 13.7 “WinWebMail 软件产品最终用户许可协议”对话框

（3）弹出如图 13.8 所示 Destination Location 对话框，从中设置安装路径，然后单击 Next 按钮。

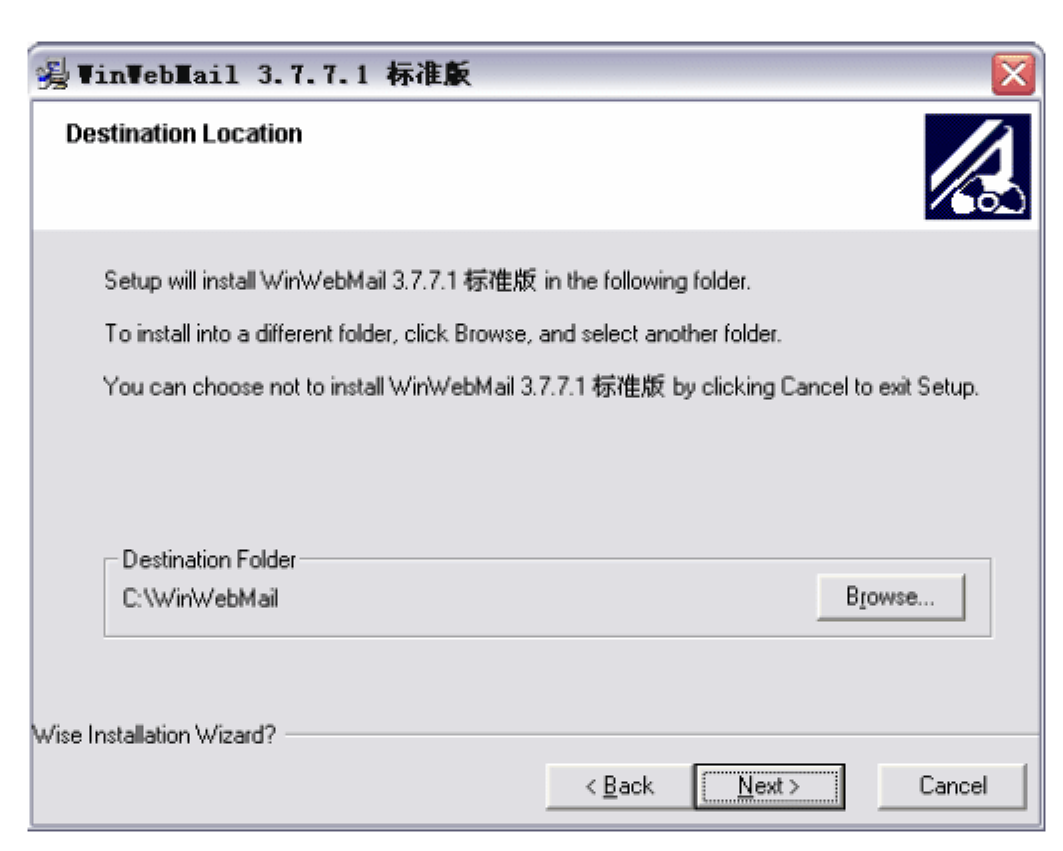


图 13.8 Destination Location 对话框



（4）弹出如图 13.9 所示 Select Program Manager Group 对话框，单击 Next 按钮。

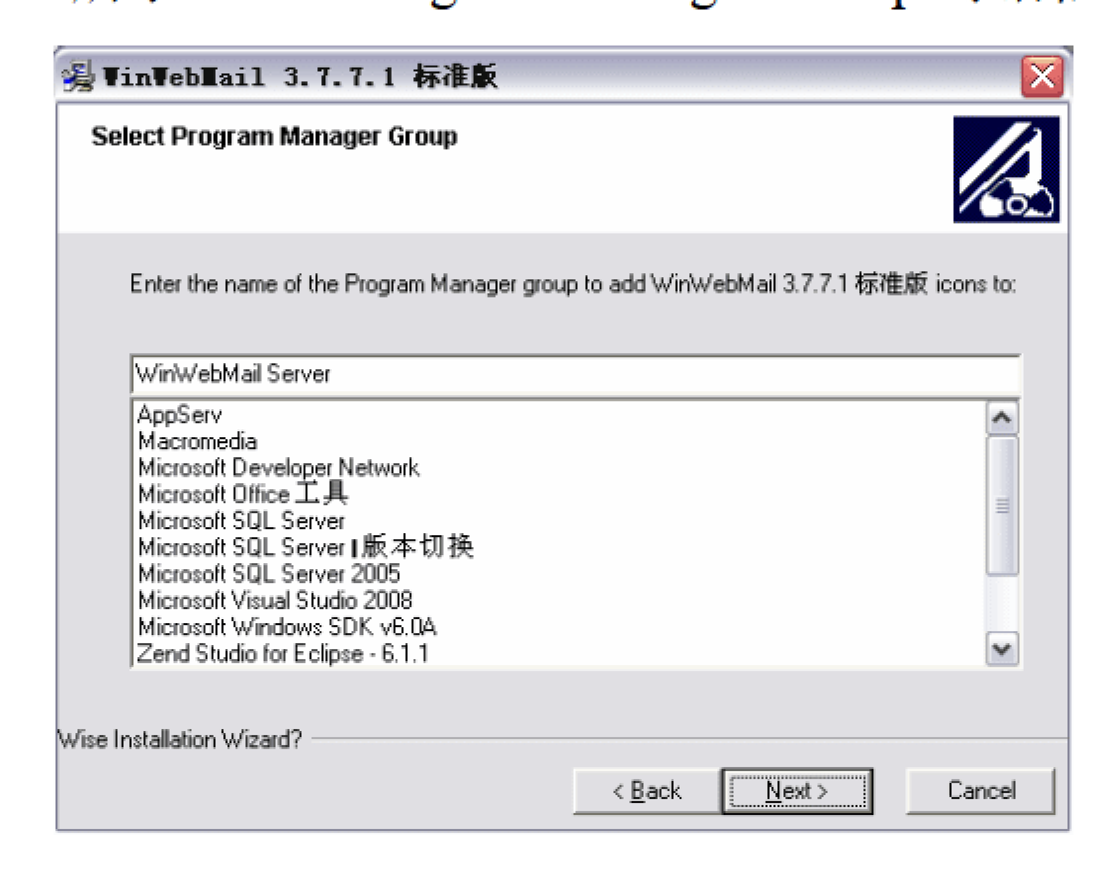


图 13.9　Select Program Manager Group 对话框

（5）弹出如图 13.10 所示对话框，单击 Finish 按钮，完成 Winwebmail 3.7.7.1 标准版的安装。

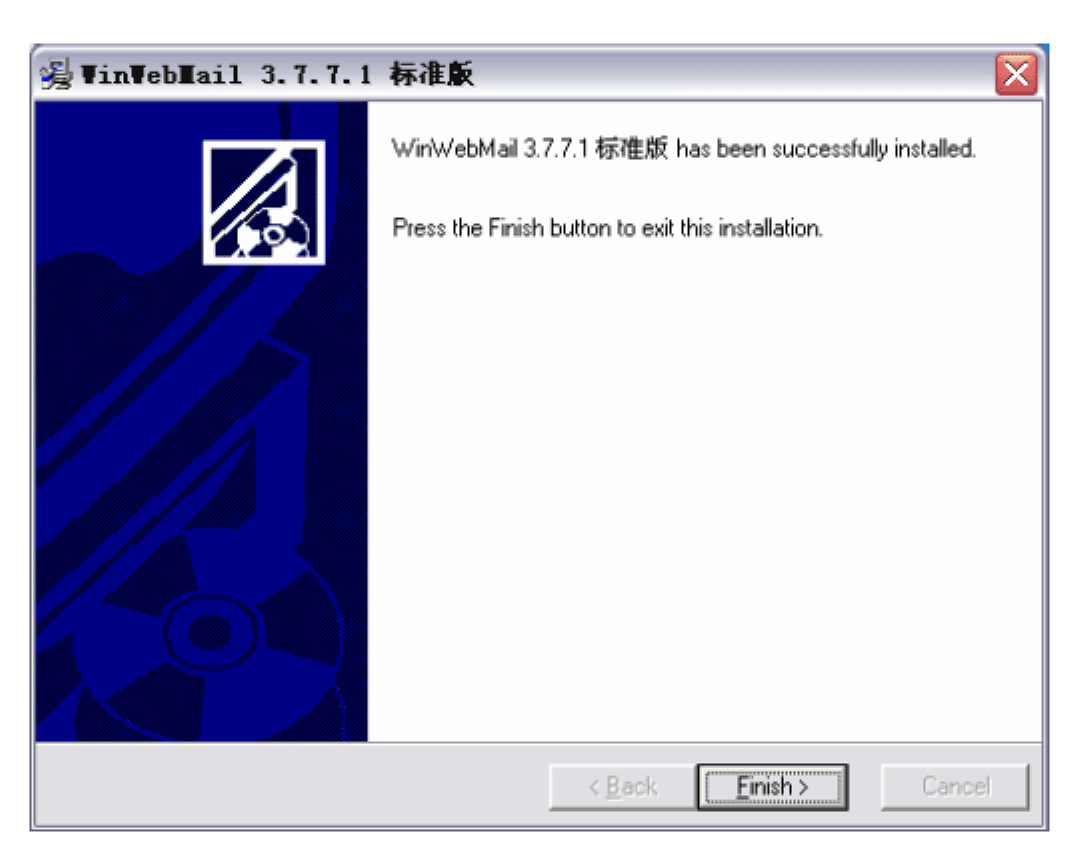


图 13.10　完成 WinWebMail 3.7.7.1 标准版的安装

（6）启动 WinWebMail，此时可以在其界面右下角看到一个带红色×的蓝色图标，右击该图标，在弹出的快捷菜单中选择“服务”命令，如图 13.11 所示。

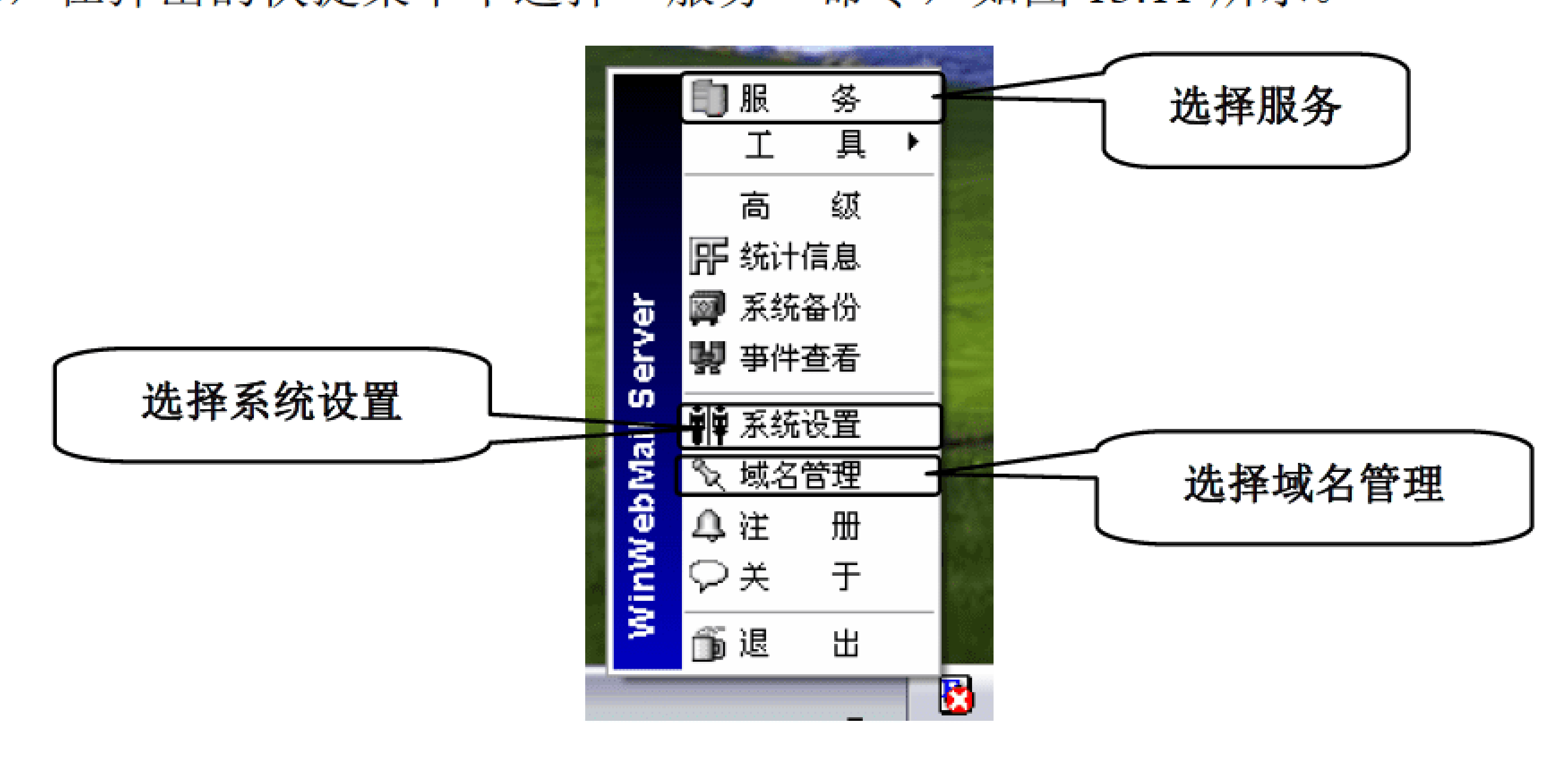


图 13.11　在快捷菜单中选择“服务”命令





Note

（7）进入如图 13.12 所示的“WinWebMail 服务”界面，选中“修改”复选框，输入 DNS 的 IP 地址，单击“启动 WinWebMail 服务程序”按钮，然后单击✔（绿色对号）按钮，完成服务选项配置。

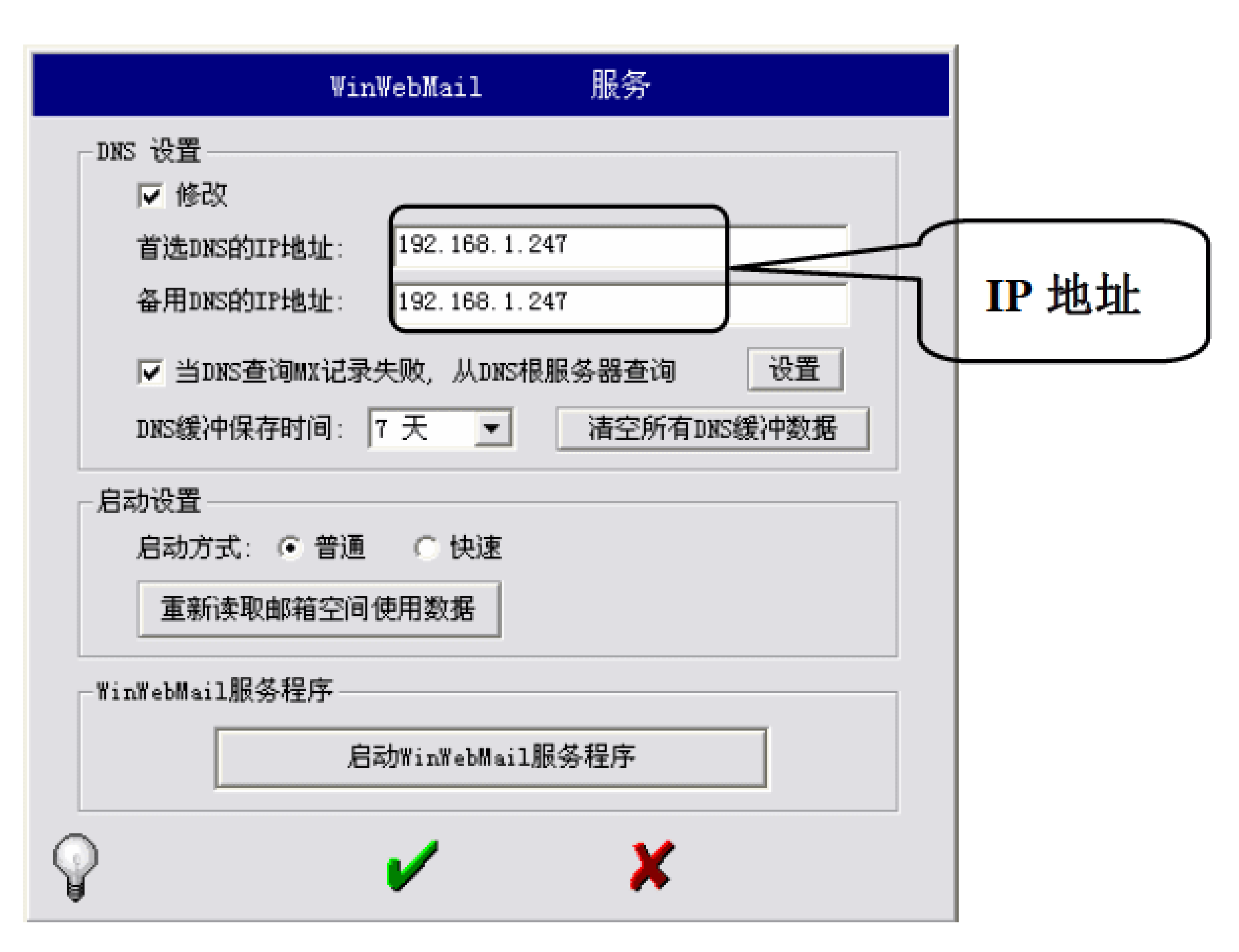


图 13.12　WinWebMail 服务

（8）重新进入如图 13.11 所示界面，右击右下角的图标，在弹出的快捷菜单中选择“域名管理”命令，弹出如图 13.13 所示界面。在该界面中添加域名，然后单击✔（绿色对号）按钮。

（9）重新进入如图 13.11 所示界面，右击右下角的图标，在弹出的快捷菜单中选择“系统设置”命令，在弹出的界面中依次添加用户名、密码，选择所在域名，然后单击✔（绿色对号）按钮，如图 13.14 所示。至此，WinWebMail 配置完毕。

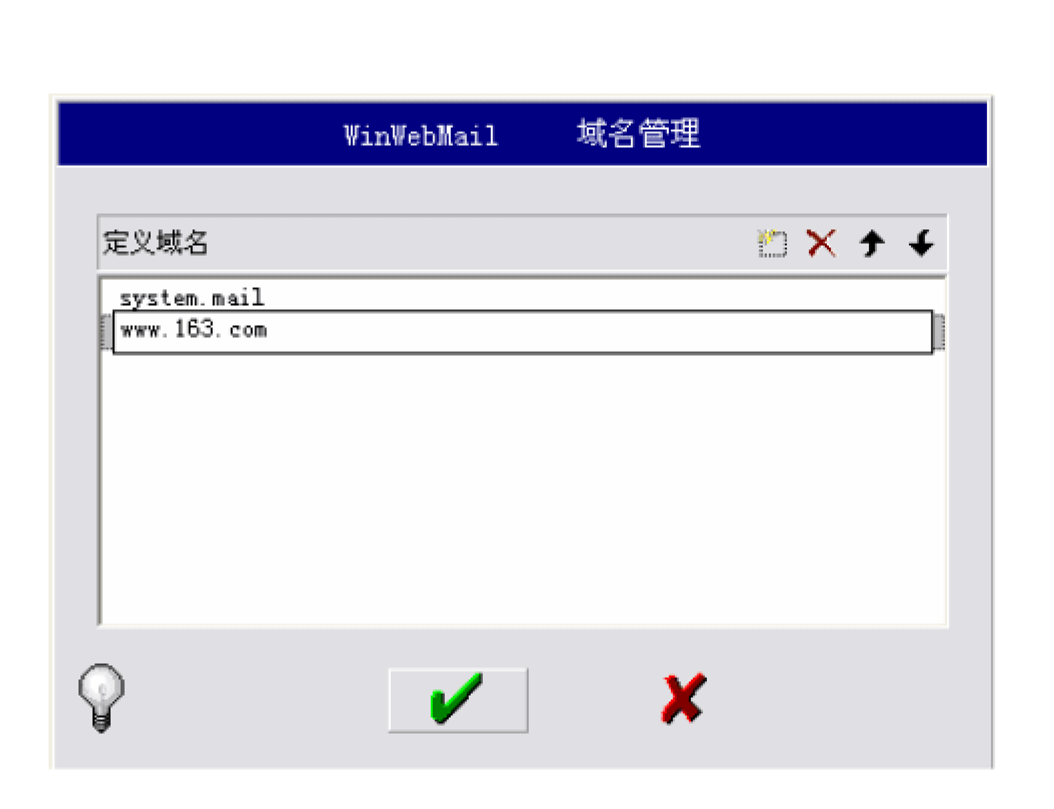
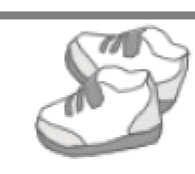
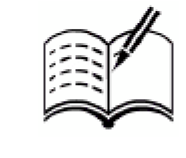


图 13.13　WinWebMail 域名管理



图 13.14　WinWebMail 系统设置





13.4　登录邮件服务器

用户登录邮件服务器时，在如图 13.15 所示的登录界面中输入 POP3 服务器地址、用户在 POP3 服务器注册的邮箱名称和密码，单击“登录”按钮，将数据提交到 chklogin.php 文件中，完成与服务器的连接。

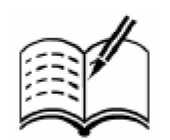


图 13.15　登录邮件服务器

在 chklogin.php 文件中，获取表单提交的服务器地址、邮箱名称和密码，调用 POP3 类中的 open 和 login 方法，完成与服务器的连接，并且将登录成功后的服务器地址、邮箱名称和密码赋给指定的 SESSION 变量。其关键代码如下：

```
<?php
session_start();
include("mailclass.php");                    //包含发邮件类
$hostname=$_POST['servername'];              //获取提交的服务器地址
$mailname=substr($_POST['mailname'],0,strpos($_POST['mailname'],'@'));    //获取提交的用户邮箱地址
$mailpwd=$_POST['mailpwd'];                  //获取提交的用户邮箱密码
$rec=new pop3($hostname,110,10);             //对发邮件类进行初始化
if(!$rec->open()){     //如果调用该类的 open 方法返回 false 值，则说明没有成功连接上服务器，这时给出提示
    echo"<script>alert('对不起，无法连接服务器！');history.back();</script>";
    exit;
}
//如果调用发邮件类的 login 方法返回 false 值，则说明用户输入的名称或密码错误
if(!$rec->login($mailname,$mailpwd)){
    echo"<script>alert('对不起，您的用户名或密码输入错误！');history.back();</script>";
    exit;
}
$_SESSION['host']=$hostname;//如果用户登录成功，则注册 SESSION 变量 host 保存服务器地址以备其他页面使用
//如果用户登录成功，则注册 SESSION 变量 user 保存用户邮箱地址以备其他页面使用
$_SESSION['user']=$_POST['mailname'];
$_SESSION['pass']=$mailpwd; //如果用户登录成功，则注册 SESSION 变量 pass 保存用户邮箱密码以备其他页面使用
$rec->close(); //调用发邮件类的 close 方法关闭与 POP3 服务器的连接
header("location:list.php");    //定位到发件箱页面
?>
```





13.5 邮件收发

13.5.1 邮件收发模块概述

成功登录邮件服务器之后，就可以完成邮件的发送、接收、查看、查找和删除了。邮件发送系统提供的功能如图 13.16 所示。

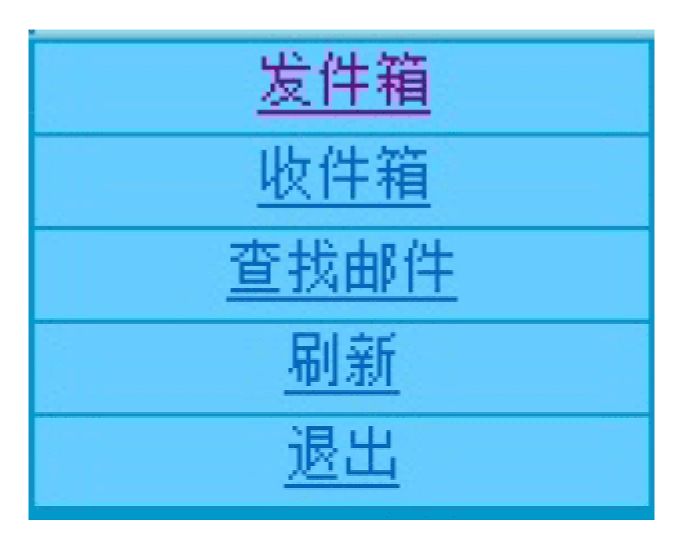


图 13.16 邮件发送系统的功能

在"发件箱"中可以完成无附件邮件和带附件邮件的发送操作；在"收件箱"中可以查看接收到的邮件主题，以及邮件的具体内容，同时还可以完成附件的下载操作；在"查找邮件"中可以根据邮件主题和发件人对邮件进行查找；"刷新"功能用于完成对页面的刷新操作，以此显示邮箱最新收到的邮件信息；"退出"用于实现当前登录用户的退出。

13.5.2 发送邮件

发送邮件的操作由 list.php 和 sendmail.php 两个文件完成。在 list.php 中定义发送邮件表单，如图 13.17 所示。

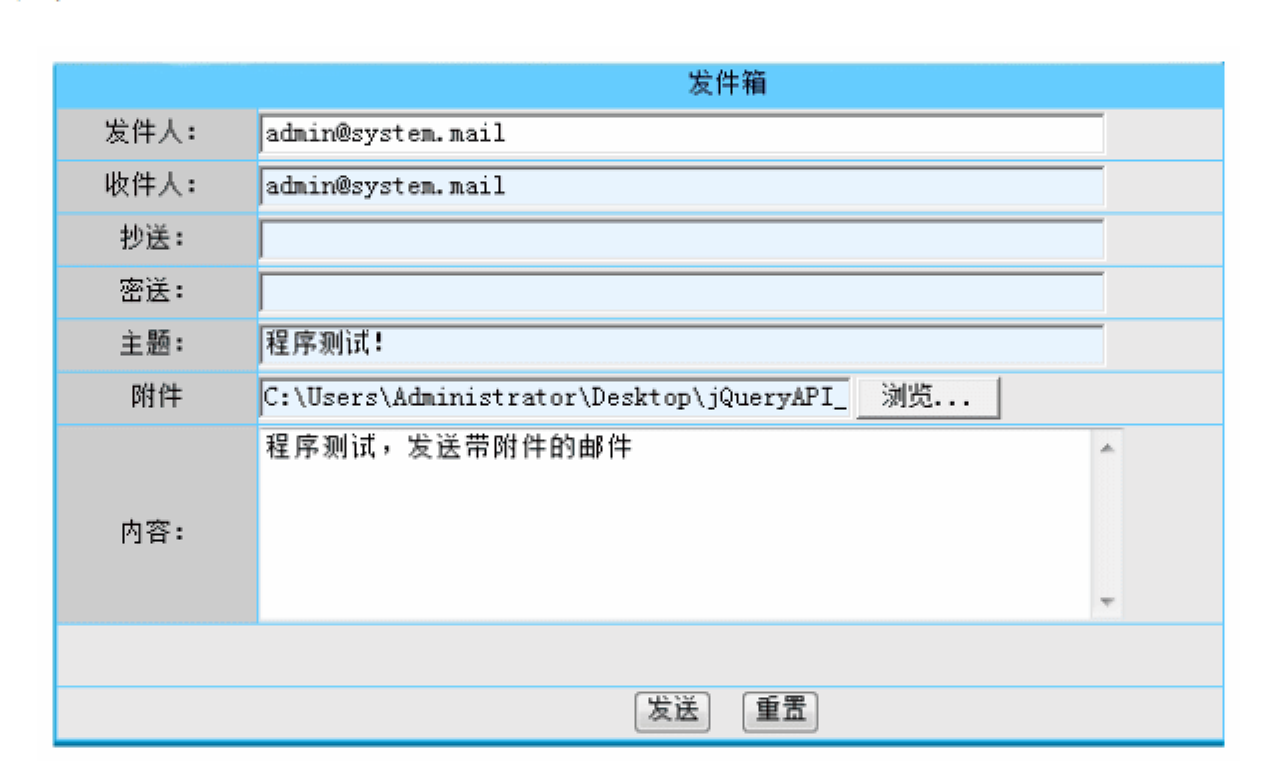


图 13.17 发送邮件

为了缓解服务器压力，在添加发送邮件的信息时，应用 JavaScript 脚本对表单提交的数据进行了验证，其中发件人、收件人和邮件内容为必填内容，并且格式必须正确，而附件为可选项，抄送和密送不必填写。验证表单提交数据格式是否正确使用的是 JavaScript 脚本的自定义函数 chkinput()，该函数的部分代码如下：

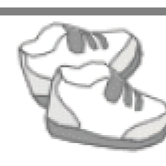





```
<!--******************************list.php*********************************-->
<script language="JavaScript">
 function chkinput(form)                    //定义 chkinput()函数，用来验证用户输入的信息
   {
     if(form.mailfrom.value=="")            //判断用户输入的信息是否为空
       {
//如果用户输入的信息为空，则利用 window 的 alert()方法弹出一个提示框，告诉用户某些必添的内容没有填写
         alert("请输入发件人地址!");
         form.mailfrom.select();            //选择当前窗体元素
         return(false);                     //返回 false 值使表单不能提交
       }
      …                                       //省略了功能类似的代码，详见配套资源
//由于电子邮箱中一定含有字符@，所以通过用户输入的邮箱地址是否含有字符@就可以判断用户输入的邮箱地址是否正确，函数 indexOf()的作用是判断该函数参数在某字符串中第一次出现的位置，如果结果小于 0 则说明没有查找到该字符
       if(form.mailfrom.value.indexOf('@')<0)
       {
         alert("发件人 E-mail 格式输入错误!");
         form.mailfrom.select();
         return(false);
       }
      …
       return(true);          //如果用户输入的信息通过验证，则该函数返回 true，从而使表单能够提交
   }
</script>
```

在 sendmail.php 文件中，首先获取表单提交的邮件内容，然后对邮件标题和内容进行编码格式转换，最后根据附件的内容进行判断，如果附件为空，那么直接应用 mail()函数完成邮件的发送；如果附件不为空，则需要通过文件系统函数读取附件的内容，对其编码格式进行转换，定义到邮件体中后，应用 mail()函数完成邮件的发送。其关键代码如下：

```
<!—**************************sendmail.php*******************************-->
<?php
include("function.php");
$mailfrom=$_POST['mailfrom'];                              //获取发件人地址
$mailto=$_POST['mailto'];                                  //获取收件人地址
$mailcc=$_POST['mailcc'];                                  //获取抄送人地址
$mailbcc=$_POST['mailbcc'];                                //获取密送人地址
$mailsubject=base64_encode($_POST['mailsubject']);         //获取邮件主题
$mailbody = base64_encode($_POST['mailbody']);             //获取邮件内容
if(empty($_FILES['upload_file']['name'])){
     $headers = "From:$mailfrom \r\n Cc:$mailcc \r\n Bcc:$mailbcc";    //定义邮件头
     $headers .= "Reply-To:$mailfrom\r\n";
     $headers .= "Content-type: text/html;\r\n charset=iso-8859-1 \r\n";
     if(@mail($mailto, $mailsubject,$mailbody,$headers)){ //通过 mail()函数发送邮件，并给出邮件发送结果
          echo "<script>alert('邮件发送成功！ ');history.back()</script>";
```




Note

```
        }else{
            echo "<script>alert('邮件发送失败！ ');history.back()</script>";
        }
    }else{
        $file = $_FILES['upload_file']['tmp_name'];          //获取附件名称
        $boundary = uniqid("");                              //定义分割线
        $headers = "MIME-Version: 1.0 \n";
        $headers .= "Content-type: multipart/mixed; Boundary= $boundary \r\n";
        $headers .= "From:$mailfrom\r\n";                    //定义邮件头
        if($_FILES['upload_file']['type']){                  //判断附件类型
            $mimeType = $_FILES['upload_file']['type'];
        }else{
            $mimeType ="application/unknown";                //获取附件类型
        }
        $fileName = $_FILES['upload_file']['name'];          //获得附件名称
        $fp = fopen($file, "r");    //打开文件
        $read = fread($fp, filesize($file));                 //将附件读入变量$read
        $read = base64_encode($read);                        //转换 base64 编码
        $read = chunk_split($read);          //将长字符串切成由每行 76 个字符组成的小块
        /*定义邮件体*/
        $body = "--$boundary--
    Content-type: text/plain; charset=iso-8859-1
    $mailbody
    --$boundary--
    Content-type: $mimeType; name=$fileName
    Content-disposition: attachment; filename=$fileName
    Content-transfer-encoding: base64
    $read
    --$boundary--";
        if(mail($mailto, $mailsubject,$body,$headers)){ //用 mail()函数发送邮件
            echo "<script>alert('附件发送成功！ ');history.back()</script>";
        }else{
            echo "<script>alert('附件发送失败！ ');history.back()</script>";
        }
    }
    ?>
```

说明：

当要发送的邮件中带有附件时，首先需要将邮件的 MIME 的 Content-type 类型定义成 multipart/mixed，然后定义邮件内容分割线（一串字符或数字），最后将文件内容编写在邮件主体中进行发送。

注意：

PHP 的 mail()函数只能将邮件发送到 SMTP 的邮件转发器中，然后通过转发器将邮件发送出去。



13.5.3　接收邮件

接收邮件的核心是调用 pop3 类中的 open、login、stat 和 listmail 方法，实现邮件的接收操作。pop3 类独立存储于 mailclass.php 中，将在 13.7 节中进行详细讲解。这里将具体介绍系统是如何调用 pop3 类实现接收邮件功能的。收件箱页面如图 13.18 所示。

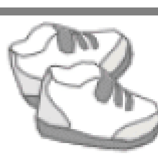


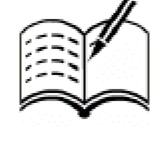
图 13.18　收件箱页面

在 receivemail.php 文件中，首先初始化 SESSION 变量，判断当前用户是否具有访问权限；然后通过 include 语句载入 pop3 类文件 mailclass.php；最后，实例化 pop3 类，通过返回的对象调用类中的方法，循环输出收件箱中的邮件信息。其关键代码如下：

```
<!--************************receivemail.php***************************-->
<?php
session_start();
if(!isset($_SESSION['user'])){
        echo "<script>alert('请不要非法登录本站！');history.back();</script>";
        exit;
}       //以上代码通过判断 SESSION 变量 user 的值是否为空来验证用户是否非法登录
include("mailclass.php");                          //包含发邮件类
$rec=new pop3($_SESSION['host'],110,10);           //对发邮件类进行实例化
$rec->open();                  //调用发邮件类的 open 方法来与 POP3 服务器建立连接
//调用发邮件类的 login 方法来验证用户身份
$rec-> login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION['pass']);
$rec->stat();                  //调用发邮件类的 stat 方法来获得邮件总数和邮件总字节数
$rec->listmail();              //调用发邮件类的 listmail 方法获得每个邮件的大小及序号
?>……
<?php
if($rec->messages<=0){         //判断邮件记录数
?>
<table width="650" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
        <td bgcolor="#66CCFF"><div align="center">您的收件箱内无邮件！</div></td>
        </tr>
</table>
<?php
}else{                         //如果邮件存在，则执行下面的操作
?>
```

Note





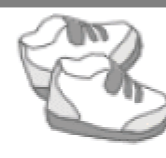

```
<table width="650" border="0" align="center" cellpadding="0" cellspacing="0">
      <tr>
       <td height="40"   bgcolor="#66CCFF">
<table width="650" height="40" border="0" cellpadding="0" cellspacing="1">
       <tr>
       <td height="20" colspan="2" bgcolor="E9E8E8"><div align="center">主题</div></td>
            <td width="136" bgcolor="E9E8E8"><div align="center">发件人</div></td>
            <td width="183" bgcolor="E9E8E8"><div align="center">时间</div></td>
            <td width="95" bgcolor="E9E8E8"><div align="center">大小</div></td>
       </tr>
<?php
       for($i=1;$i<=$rec->messages;$i++){    //循环输出邮件数据
              $rec->getmail($i);
?>
       <tr>
              <td width="43" height="20" bgcolor="E9E8E8"><div align="center">
                     <a   href="deletemail.php?del=<?php   echo   $i;?>"><img   src="images/drop.png" border="0"></a>
              </div></td>
              <td width="187" bgcolor="E9E8E8"><div align="left">
                     <a href="readmail.php?mailid=<?php echo $i;?>">
<?php
              for($j=0;$j<count($rec->head);$j++){  //通过循环提取所有保存在 head 数组中的邮件头的信息
                     if(substr($rec->head[$j],0,20)=="Subject: =?GB2312?Q?"){ //获取邮件主题
                            echo quoted_printable_decode(substr(htmlspecialchars($rec->head[$j]),20,
strlen(trim(htmlspecialchars($rec->head[$j])))-22))."<br>\n";          //输出邮件主题
                     }else if(substr($rec->head[$j],0,7)=="Subject"){    //去掉邮件主题的提示文字"Subject:"
                            echo base64_decode(substr(htmlspecialchars($rec->head[$j]),8,
strlen(trim(htmlspecialchars($rec->head[$j])))-8))."<br>\n";
                     }
              }
?>
                     </a>
       </div></td>
            <td height="20" bgcolor="E9E8E8"><div align="center">
<?php
              for($j=0;$j<count($rec->head);$j++){
                     if(substr($rec->head[$j],0,4)=="From"){       //获取发件人地址
              //去掉发件人地址的提示文字"From:"
       echo        substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5)."<br>\n";
                     }
              }
?>
              </div></td>
            <td height="20" bgcolor="E9E8E8"><div align="center">
```





Note

```
<?php
			for($j=0;$j<count($rec->head);$j++){		//获取头文件数据
				if(substr($rec->head[$j],0,4)=="Date"){ //截取时间数据
	echo	substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-10)."<br>\n";//输出时间
				}
			}
?>
			</div></td>
		<td height="20" bgcolor="E9E8E8"><div align="center">
<?php
				$size=$rec->mail_list[$i]['size'];		//获取文件大小
				if($size>=1024){			//计算文件大小，以不同单位进行换算输出
					echo substr(($size/1024),0,4)."M";
				}elseif($size>1024*1024){
					echo substr(($size/(1024*1024)),0,4)."G";
				}elseif($size<1024){
					echo ($size)."KB";
				}
?>
			</div></td>
		</tr>
<?php
			$rec->head=NULL;
		}
?>
</table>
			</td>
		</tr>
</table>
<table width="650" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
		<tr>
			<td width="114"></td>
<td width="536"><div align="right">您的邮箱中共有邮件: <?php echo $rec->messages;?> 封占用空间:
<?php
$totalsize=$rec->size;			//计算邮件服务器空间大小
if($totalsize>=1024){
		echo substr(($totalsize/1024),0,4)."M";
}elseif($totalsize>1024*1024){
		echo substr(($totalsize/(1024*1024)),0,4)."G";
}elseif($totalsize<1024){
		echo ($totalsize)."KB";
}
?>
			 </div></td>
		</tr>
</table>
<?php
```



```
$rec->close();                    //关闭服务器
}
?>
```



Note

13.5.4 查看邮件详细内容

在收件箱页面（receivemail.php 文件）中，单击邮件主题超链接，将跳转到 readmail.php 页中。在该文件中将根据超链接传递的邮件 ID 值，输出指定邮件的详细信息。其仍然是调用 pop3 类中的方法，完成邮件详细信息的输出操作。其关键代码如下：

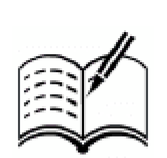
```
<?php
session_start();                        //初始化 SESSION 变量
if(!isset($_SESSION['user'])){          //判断用户的权限
    echo "<script>alert('请不要非法登录本站！');history.back();</script>";
    exit;
}
$mailid=$_GET['mailid'];                //获取超链接传递的邮件 ID 值
include("mailclass.php");               //载入 pop3 类
$rec=new pop3($_SESSION['host'],110,10);        //实例化 pop3 类
$rec->open();                                   //连接邮件服务器
$rec->login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION['pass']); //用户登录
$rec->stat();                           //验证用户是否正确登录
$rec->listmail();                       //读取邮件
$rec->getmail($mailid);                 //根据超链接传递的 ID 值，获取指定邮件的内容
?>
<table width="650" border="0" align="center" cellpadding="0" cellspacing="1">
    <tr>
    <td width="152" height="25" bgcolor="#CCCCCC"><div align="center">发 件 人:</div></td>
        <td colspan="2" bgcolor="#E9E8E8"><div align="left">
<?php
for($j=0;$j<count($rec->head);$j++){                    //获取发件人信息
    if(substr($rec->head[$j],0,4)=="From"){             //输出发件人
    echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j]))-5)."<br>\n";
    }
}
?>
          </div></td>
    </tr>
   <tr>
    <td height="25" bgcolor="#CCCCCC"><div align="center">收件人:</div></td>
<td height="20" colspan="2" bgcolor="#E9E8E8"><div align="left"><?php echo $_SESSION['user']; ?></div></td>
    </tr>
   <tr>
    <td height="25" bgcolor="#CCCCCC"><div align="center">邮件主题:</div></td>
```






```
                <td height="20" colspan="2" bgcolor="#E9E8E8"><div align="left">
<?php
for($j=0;$j<count($rec->head);$j++){                                    //获取邮件主题
        if(substr($rec->head[$j],0,20)=="Subject: =?GB2312?Q?"){    //对邮件主题进行截取
                echo quoted_printable_decode(substr(htmlspecialchars($rec->head[$j]),20,
strlen(trim(htmlspecialchars($rec->head[$j])))-22))."<br>\n";            //输出邮件主题
        }else if(substr($rec->head[$j],0,7)=="Subject"){
                echo base64_decode(substr(htmlspecialchars($rec->head[$j]),8,
strlen(trim(htmlspecialchars($rec->head[$j])))-8))."<br>\n";
        }
}
?>
                </div></td>
        </tr>
        <tr>
        <td height="25" bgcolor="#CCCCCC"><div align="center">发件时间:</div></td>
                <td height="20" colspan="2" bgcolor="#E9E8E8"><div align="left">
<?php
for($j=0;$j<count($rec->head);$j++){            //获取发件时间
        if(substr($rec->head[$j],0,4)=="Date"){ //对发件时间数据进行截取输出
echo    substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->    head[$j])))-10)."<br>\n";
        }
}
?>
                </div></td>
        </tr>
        <tr>
        <td height="25" bgcolor="#cccccc"><div align="center">邮件大小:</div></td>
                <td height="20" bgcolor="#E9E8E8"><div align="left">
<?php
                        $size=$rec->mail_list[$mailid]['size'];    //获取邮件大小
                        if($size>=1024){                            //对邮件大小进行换算输出
                                echo ($size/1024)."M";
                        }elseif($size>1024*1024){
                                echo ($size/(1024*1024))."G";
                        }elseif($size<1024){
                                echo ($size)."KB";
                        }
                        if(isset($rec->body[6])){                   //判断是否存在附件
                                $file='';
                                $filename=substr(strrchr($rec->body[6],"="),1,200);//获取附件名称
                                for($i=0;$i<count($rec->body);$i++){            //读取附件数据
                                        if($i>7 && $i<count($rec->body)-2){
                                                $file.=base64_decode($rec->body[$i]);//对附件进行编码转换
                                        }
                                }
                                file_put_contents("file/".$filename,$file);     //将附件数据写入到指定文件中
```






```
                    }
?>
            </div></td>
          <td height="20" bgcolor="#E9E8E8"><div align="center">
            <input type="button" value="删除" class="buttoncss"
onClick="javascript:window.location='deletemail.php?del=<?php echo $mailid;?>&filename=<?php echo $filename;?>'">
            </div></td>
      </tr>
<?php
      if(isset($rec->body[6])){          //判断附件是否存在，存在则创建附件下载超链接
?>
     <tr>
      <td height="25" bgcolor="#cccccc"><div align="center">附件:</div></td>
          <td width="352" height="20" bgcolor="#E9E8E8"><?php echo '
<a href="down.php?filename='.$filename.'" target="_blank">'.$filename.'</a>'; ?>
            </td>
      <td width="142" bgcolor="#E9E8E8"> </td>
      </tr>
<?php
      }
?>
      <tr>
      <td height="325" colspan="3" bgcolor="#E9E8E8" valign="top">
<?php
                  if(isset($rec->body[6])){
                        echo base64_decode($rec->body[3])."<br>";
                  }else{
                        if(substr($rec->body[1],0,1)!="="){
                              echo base64_decode($rec->body[1]);
                        }else{
                              echo quoted_printable_decode($rec->body[1]);
                        }
                  }
?>
            </td>
      </tr>
</table>
```

13.5.5 下载附件

在查看邮件详细内容页面中单击附件超链接，将链接到 down.php 文件，在该文件中根据超链接传递的附件名称实现附件的下载。其运行效果如图 13.19 所示。



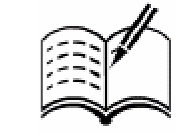




图 13.19　下载附件

在 down.php 文件中，根据超链接传递的附件名称，通过 header()函数完成附件的下载操作。其关键代码如下：

```
<?php
session_start();                                //初始化 SESSION 变量
$filename='file/'.$_GET['filename'];            //获取附件在服务器中的存储位置
if(!isset($_SESSION['user'])){                  //判断当前用户的访问权限
    echo "<script>alert('对不起，本站暂时停止该文件下载!');history.back();</script>";
    exit;
}
$fp=fopen($filename,"r");                       //打开指定的文件
header("Content-type:application/octet-stream");
header("Accept-ranges:bytes");
header("Accept-length:".filesize($filename));
header("Content-Disposition:attachment;filename=".$filename);
echo fread($fp,filesize($filename));            //输出文件
fclose($fp);                                //关闭文件
?>
```

13.5.6　删除邮件

在查看邮件详细内容页面中，单击“删除”按钮，将跳转到 deletemail.php 文件中，根据超链接传递的邮件 ID 和附件名称，调用 pop3 类中的 dele 方法完成邮件的删除操作。如果存在附件，则应用 unlink()函数删除存储于服务器下的附件。deletemail.php 文件的代码如下：

```
<!—*************************deletemail.php*****************************-->
<?php
session_start();
if(isset($_SESSION['user'])){                           //判断当前用户的权限
    include("mailclass.php");                           //包含 pop3 发邮件类
    $rec=new pop3($_SESSION['host'],110,10);            //对类进行初始化
```

```
        $rec->open();                             //连接 POP3 服务器
        //对当前用户进行身份验证
        $rec->login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION ['pass']);
        $rec->dele($_GET['del']);    //调用 pop3 类的 dele 方法来删除$_GET[del]指定的邮件
        $rec->close();               //关闭与 POP3 服务器的连接，并执行删除操作
        if(!empty($_GET['filename'])){
              unlink('file/'.$_GET['filename']);
        }
        echo "<script>alert('删除成功!');window.location.href='receivemail.php';</script>";
    }else{
        echo  "<script>alert('您不具备删除权限，请正确登录!');window.location.href='index.php';;</script>";
    }
    ?>
```

13.5.7　查找邮件

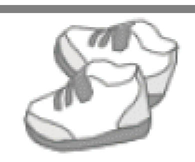
为了方便用户，该邮件发送系统提供了两种搜索方式，分别为按邮件主题进行查找和按发件人地址进行查找。查找邮件的操作在 findmail.php 文件中完成，页面运行效果如图 13.20 所示。



图 13.20　查找邮件

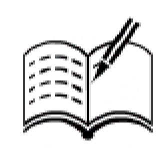
查找邮件创建的 form 表单如下：

```
    <form name="form1" method="post" action="<?php echo $PHP_SELF;?>" onSubmit="return chkinput(this)">
        <tr>
        <td width="143"><div align="right">查找邮件:</div></td>
            <td width="92"><div align="center">
              <select name="findtype" class="inputcss">
                    <option selected value=1>邮件主题</option>
                      <option value=0>发件人</option>
                    </select>
              </div></td>
            <td width="191"><div align="center">
                    <input  type="text"  name="info"  size="30"  class="inputcss"  style="background-color:#e8f4ff "
    onMouseOver="this.style.backgroundColor='#ffffff'" onMouseOut="this.style.backgroundColor='#e8f4ff'">
              </div></td>
            <td width="224"><div align="left">
                      <input type="submit" value="开始" class="buttoncss" name="submit">
              </div></td>
```



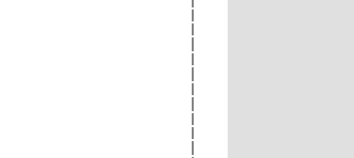
```
		</tr>
	</form>
```

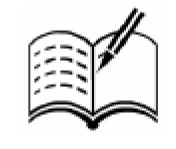
查找邮件的处理在 findmail.php 文件中完成，其关键代码如下。




```
<!--****************************findmail.php********************************-->
…
<?php
if(isset($_POST['submit'])){                    //判断提交按钮值是否被设置
?>
	<tr>
	<td width="188" height="20" bgcolor="#E9E8E8"><div align="center">主题</div> </td>
		<td width="153" height="20" bgcolor="#E9E8E8"><div align="center">发件人</div> </td>
		<td width="209" height="20" bgcolor="#E9E8E8"><div align="center">时间</div> </td>
		<td width="97" height="20" bgcolor="#E9E8E8"><div align="center">大小</div> </td>
	</tr>
<?php
include("mailclass.php");                       //载入类文件
$rec=new pop3($_SESSION['host'],110,10);        //实例化类
$rec->open();                                   //连接服务器
$rec->login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION['pass']); //用户登录
$rec->stat();                    //调用 pop3 类的 stat 方法取得总邮件数和总邮件大小
$rec->listmail();                //调用 pop3 类的 listmail 方法取得每个邮件的大小及序号
if($_POST['findtype']==1){       //判断查询的类型，1 表示根据邮件主题查询
	$sum=0;                                     //用$sum 记录查找到邮件的总个数
	for($i=1;$i<=$rec->messages;$i++){          //通过循环遍历所有邮件
		$rec->getmail($i);                      //取得第$i 封邮件头及邮件内容
		for($j=0;$j<count($rec->head);$j++){    //通过循环获取邮件头所有内容
			if(substr($rec->head[$j],0,7)=="Subject"){//获取邮件主题
			//去掉邮件主题的提示文字"Subject:"
$head1=substr(htmlspecialchars($rec->head[$j]),8,strlen(trim(htmlspecialchars($rec->head[$j])))-8);
			}
		}
		if(preg_match("/$_POST[info]/i",trim($head1))){//利用正则表达式进行匹配查找
			$sum++; //如果查找到匹配的主题，则使 sum 值加 1
?>
	<tr>
	<td height="20" bgcolor="#E9E8E8"><div align="left">
			<a href="readmail.php?mailid=<?php echo $i;?>">
<?php
for($j=0;$j<count($rec->head);$j++){   //显示邮件主题
	if(substr($rec->head[$j],0,7)=="Subject"){
echo  substr(htmlspecialchars($rec->head[$j]),8,strlen(trim(htmlspecialchars($rec->head[$j])))-8)."<br>\n";
	}
}
?>
			</a>
```





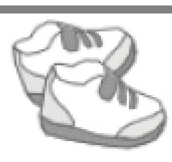
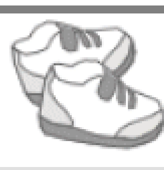



```
                </div></td>
              <td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){     //显示发件人地址
        if(substr($rec->head[$j],0,4)=="From"){
echo   substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5)."<br>\n";
        }
}
?>
                </div></td>
              <td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){     //显示发件时间
        if(substr($rec->head[$j],0,4)=="Date"){
echo   substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-10)."<br>\n";
        }
}
?>
                </div></td>
              <td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
                    $size=$rec->mail_list[$i]['size'];   //显示邮件大小
                    if($size>=1024){
                            echo substr(($size/1024),0,4)."M";
                    }elseif($size>1024*1024){
                            echo substr(($size/(1024*1024)),0,4)."G";
                    }elseif($size<1024){
                            echo ($size)."KB";
                    }
?>
                </div></td>
        </tr>
<?php
                }
                $rec->head=NULL;
        }
        if($sum==0){   //如果$sum 的值最终还为 0，则说明没有查找到匹配的邮件主题，这时给出提示
?>
        <tr>
<td height="20" colspan="4" bgcolor="#E9E8E8"><div align="center">对不起,没有查找到您要找的邮件!</div></td>
        </tr>
<?php
        }
}elseif($_POST['findtype']==0){ //如果$_POST['findtype']的值为 0，则按发件人地址进行查找
        $sum=0;
```






```
        for($i=1;$i<=$rec->messages;$i++){
            $rec->getmail($i);
            for($j=0;$j<count($rec->head);$j++){
                if(substr($rec->head[$j],0,4)=="From"){
$head1=substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5);
                }
            }
            if(trim($head1)==trim($_POST['info'])){
                $sum++;
?>
        <tr>
<td height="20" bgcolor="#E9E8E8"><div align="left"><a href="readmail.php?mailid=<?php echo $i;?>">
<?php
for($j=0;$j<count($rec->head);$j++){
        if(substr($rec->head[$j],0,7)=="Subject"){
echo  substr(htmlspecialchars($rec->head[$j]),8,strlen(trim(htmlspecialchars($rec->head[$j])))-8)."<br>\n";
        }
}
?>
            </a></div></td>
            <td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){
        if(substr($rec->head[$j],0,4)=="From"){
echo  substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5)."<br>\n";
        }
}
?>
            </div></td>
          <td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){
        if(substr($rec->head[$j],0,4)=="Date"){
echo  substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-10)."<br>\n";
        }
}
?>
            </div></td>
          <td height="2" bgcolor="#E9E8E8"><div align="center">
<?php
                $size=$rec->mail_list[$i]['size'];
                    if($size>=1024){
                        echo substr(($size/1024),0,4)."M";
                    }elseif($size>1024*1024){
```

```
                        echo substr(($size/(1024*1024)),0,4)."G";
                    }elseif($size<1024){
                        echo ($size)."KB";
                    }
?>
            </div></td>
        </tr>
<?php
            }
            $rec->head=NULL;
        }
        if($sum==0){
?>
        <tr>
<td height="20" colspan="4" bgcolor="#E9E8E8"><div align="center">对不起,没有查找到您要找的邮件!</div></td>
        </tr>
<?php
        }
}
}
?>
```

13.6 注销系统

为了保证个人隐私，用户使用完本系统后应及时退出。本邮件发送系统完成注销系统功能的文件为 logout.php，其详细代码如下：

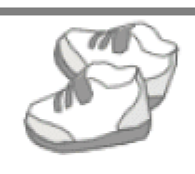
```
<!--***************************logout.php********************************-->
<?php
session_start();                                        //初始化 SESSION 变量
session_destroy();                                      //撤消所有 SESSION 变量
header("location:index.php");                           //重新定位到登录页面
?>
```

注销系统模块通过 session_destroy()函数撤消所有的 SESSION 变量，这样保存在 SESSION 变量中的所有与用户有关的信息也将不复存在。

13.7 技术提炼

13.7.1 利用 mail()函数发送电子邮件

PHP 中提供了可以直接发送电子邮件的 mail()函数。对于非商业用的邮件系统使用该函数最为方便。由于 mail()函数不能进行 SMTP 认证，所以最好安装一个 SMTP 服务器来转发邮件。mail()函数的语法如下：






```
bool mail(string to, string subject, string message[ ,string additional_headers][ ,string additional_parameters])
```

参数说明如表 13.1 所示。

表 13.1　mail()函数的参数说明

参　　数	描　　述
to	收件人地址。其表示形式有多种：user@example.com；user@example.com, anotheruser@example.com；user <user@example.com>；user <user@example.com>, another user <anotheruser@example.com>
subject	电子邮件的主题
message	电子邮件的内容
additional_headers	可选参数，用来将一些信息插入到E-mail的头部，如发送人地址、抄送地址、密送地址等。该参数如果包含多项信息，各信息之间用“\r\n”分隔
additional_parameters	其他邮件参数

注意：

使用 mail()函数发送邮件，需要在 php.ini 中对 SMTP 邮件服务器的地址和端口进行设置，否则邮件无法通过 SMTP 服务器发送出去。设置如下：

```
[mail function]
; For Win32 only.
SMTP = 192.168.1.42
smtp_port = 25
```

其中，SMTP 是发送邮件的服务器地址；smtp_port 为 SMTP 服务器的端口号，一般默认为 25。

邮件发送系统是通过 mail()函数将电子邮件发送出去的，所以应该在系统中架设一个 SMTP 服务器。该邮件发送系统利用 mail()函数发送邮件的过程如下：

```
...
if(@mail($mailto, $mailsubject,$mailbody,$headers)) {            //通过 mail()函数发送邮件，并给出
邮件发送结果
    echo "<script>alert('邮件发送成功！');history.back()</script>";
} else {
        echo "<script>alert('邮件未发送成功！');history.back()</script>";
}
```

参数说明如下。

☑　$mailto：收件人地址。

☑　$mailsubject：邮件主题。

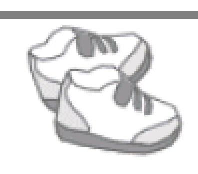
☑　$mailbody：邮件内容。

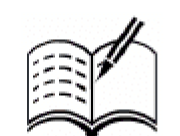
☑　$headers：邮件头内容。

如果该邮件发送成功或失败系统将给出提示，并且用“@”屏蔽错误。

13.7.2　带附件的邮件发送

发送带附件的邮件并不是简单地将附件内容上传到 POP3 服务器的某个目录下，而是





Note

首先将附件进行二进制编码，然后将该编码连接到邮件内容之后。当然，邮件内容和附件编码之间需要用标记进行分割，该标记在邮件头中定义。带附件的邮件组织形式如下：

```
Received: from mrkj ([192.168.1.227]) by MRZTS with Microsoft SMTPSVC(6.0.3790.0);
Mon, 18 Dec 2006 09:02:40 +0800
Date: Mon, 18 Dec 2006 00:57:26 +0000
Subject: 12
To: zts***@163.com
Content-type: multipart/mixed; Boundary= 4585e7760018e
From:zts***@163.com
Return-Path: zts***@163.com
Message-ID:
X-OriginalArrivalTime: 18 Dec 2006 01:02:40.0187 (UTC) FILETIME=[372680B0: 01C72240]
--4585e7760018
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: 8bit
(此处为邮件内容)
--4585e7760018
Content-type: text/plain; name=shop.txt
Content-disposition: attachment; filename=shop.txt
Content-transfer-encoding: base64
wffV4rXEtcS1xLXEtcS1xLTvtb21xNK7ICC13LXctcS1xA== （此处为附件的 base64 编码）
--4585e7760018--
.
```

带附件的邮件的不同部分之间用分割线来分隔，分割线在 Content--type 头中定义。例如，“Boundary= 4585e7760018e”，其中 4585e7760018e 表示不同部分之间的分割线。邮件的每个新部分以两个连字号（--）和分割线开始；最后一条分割线后也有两个连字号，表示这个邮件中没有其他部分了。在每条分割线后有一些行，用来告诉邮件程序这部分内容的类型。例如，上面例子中第一条分界线后面以 Content-type: text/plain 开头的行，这些行说明后面的部分是 ISO-8859-1 字符集的纯文本形式；跟在第二条分割线后的行告诉邮件程序现在的部分是一个 TXT 文件，其名称是 shop.txt。 Content-disposition 这行告诉邮件程序如果可能就以内嵌的方式显示附件，这样新的邮件程序会在消息后显示 TXT 文件的内容；如果 Content- disposition 被设置为 attachment，那么邮件程序就不会显示 TXT 文件的内容，而是显示一个连接到文件的图标，收件人如要查看附件的内容，必须单击这个图标。一般情况下，如果附件是一些文本，可将 Content-disposition 设置为 inline，这是因为现在大部分邮件程序能够不借助其他浏览器而直接显示附件（文本）的内容。如果附件不是文本（比如图片或其他类似的内容），则 Content-disposition 应设置为 attachment。

13.7.3　利用 fsockopen()函数接收电子邮件

上面介绍的 mail()函数只能发送邮件而无法接收，而 fsockopen()函数除了可以发送邮件外，还可以接收邮件。fsockopen()函数的语法格式如下：

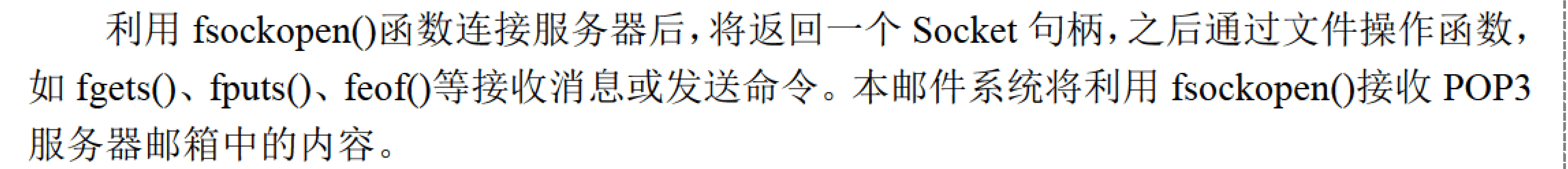
```
int fsockopen( string hostname, int port [, int errno [, string errstr [, float timeout]]])
```



其中各参数说明如表 13.2 所示。

表 13.2　fsockopen()函数的参数说明

参　　数	说　　明
hostname	在Internet中使用时，参数hostname及port分别代表网址及埠号。在UNIX下可用作IPC，hostname参数表示到socket的路径，port配置为0
port	
errno	主要用于错误处理，可以省略
errstr	
timeout	表示在指定的时间内没有连接上就中断



利用 fsockopen()函数连接服务器后，将返回一个 Socket 句柄，之后通过文件操作函数，如 fgets()、fputs()、feof()等接收消息或发送命令。本邮件系统将利用 fsockopen()接收 POP3 服务器邮箱中的内容。

说明：

fsockopen()函数可以与很多服务器建立连接，如 SMTP 服务器、POP3 服务器等。

邮件发送系统利用 fsockopen()函数连接 POP3 服务器，代码如下：

```
…
if(!$this->connection=@fsockopen($this->hostname,$this->port,&$err_no,&$err_str,$this->timeout)){
…
```

参数说明如下。

- ☑ $this->hostname：POP3 服务器地址。
- ☑ $this->port：POP3 服务器端口号。
- ☑ $err_no：系统连接 POP3 服务器发生错误时返回的错误号。
- ☑ $err_str：系统连接 POP3 服务器发生错误时返回的错误信息。
- ☑ $this->timeout：系统连接 POP3 服务器时的最大超时时间。

13.7.4　封装邮件操作类

在邮件发送系统中，将接收邮件的方法封装到 pop3 类，并且将其单独存储于 mailclass.php 中。下面将对该类进行系统讲解。

（1）定义收邮件类中的数据成员。在收邮件类 pop3 中，首先定义类中的数据成员。其代码如下：

```
<!—**************************mailclass.php******************************-->
class pop3{
        public  $hostname="";              //POP3 主机名
        public $port=110;                  //主机的 POP3 端口号，一般为 110
        public $timeout=5;                 //连接主机的最大超时时间
        public $connection=0;              //保存与主机的连接
        public $state="DISCONNECTED";//保存用户登录状态
        public $debug=0;                   //标识是否处于调试状态，如是则输出调试信息
        public $err_str='';                //如果出错，保存错误信息
        public $err_no='';                 //如果出错，保存错误号
```




```
        public $resp;           //临时保存服务器的响应信息
        public $apop;           //指示需要使用加密方式进行密码验证，一般服务器不需要
        public $messages;       //保存邮箱中邮件总数
        public $size;           //邮箱中邮件总大小
        public $mail_list;      //一维数组，保存各个邮件的大小及其在 POP 服务器上的序号
        public $head=array();   //保存邮件头内容
        public $body=array();   //保存邮件内容
```

（2）定义收邮件类的构造函数__construct()，其代码如下：

```
    <!—*************************mailclass.php****************************-->
        function __construct($server,$port=110,$time_out=5){
            $this->hostname=$server;
            $this->port=$port;
            $this->timeout=$time_out;
            return true;
        }
```

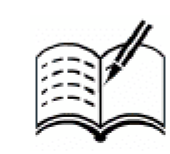
注意：

上述使用的是 PHP5 中构造函数的定义方法，而在 PHP4 中构造函数名应该与类名相同。当然，PHP5 也兼容 PHP4 中构造函数的定义方法。该构造函数的作用是为数据成员$hostname、$port、$timeout 初始化。

（3）定义 open 方法连接 POP3 服务器。在 open 方法中，通过 fsockpen()函数连接 POP3 服务器，并根据返回信息判断服务器是否连接成功，如果成功则设置标记服务器的数据成员$state 的值为 AUTHORIZATION。代码如下：

```
<!—*********************************mailclass.php***********************************-->
        function open(){
            if($this->hostname==""){          //判断服务器名是否为空，是则给出提示信息
            $this->err_str="请输入主机名！";
                return false;
            }
            if($this->debug)
            echo "正在连接服务器！";
            //利用 fsockopen()连接 POP3 服务器
            if(!$this->connection=@fsockopen($this->hostname,$this->port,&$err_no,&$err_str,$this->timeout)){
                $this->err_str="连接服务器失败，错误信息：".$err_str."错误号：".$err_no;
                return false;
            }else{
                $this->getresp();             //利用 getresp 方法处理服务器端回送的信息
                if($this->debug)              //显示返回信息
                    $this->outdebug($this->resp);
if(substr($this->resp,0,3)!="+OK"){//如果返回信息的前 3 个字符不为“+OK”说明服务器连接失败，并给出错误信息
                $this->err_str="服务器回送无效信息；".$this->resp."请检查服务器是否正确!";
```






```
                        return false;
                    }
                    $this->state="AUTHORIZATION";
                    return true;
            }
    }
```

技巧：

收邮件类判断某条命令发送是否成功的方法是：通过 substr()函数提取出返回信息的前 3 个字符，如果该字符为“+OK”表示命令被成功执行。这是因为如果 POP3 中某条命令被成功执行，返回信息的前 3 个字符为“+OK”。下面是 fsockopen()函数连接成功时返回的信息：

```
+OK Microsoft Windows POP3 Service Version 1.0 <17073156@MRZTS> ready.
```

（4）使用 getresp 方法向 POP3 服务器接收命令。getresp 用于取得服务器端回送的信息，并对信息进行如下处理：去掉最后的回车换行符后，将回送信息保存到数据成员$resp 中。getresp 方法代码如下：

```
<!—**************************mailclass.php****************************-->
    function getresp(){
            for($this->resp="";;){
                    if(feof($this->connection))
                    return false;
                    $this->resp.=fgets($this->connection,100);     //用 fgets()函数接收命令
                    $length=strlen($this->resp);                   //获取返回信息的长度
                    if($length>=2 && substr($this->resp,$length-2,2)=="\r\n"){
                            $this->resp=strtok($this->resp,"\r\n");    //去掉回车换行符
                            return true;
                    }
            }
    }
```

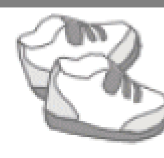
（5）使用 outdebug()方法对特殊字符进行转换。outdebug 方法的作用是将调试信息$messages 显示出来，并把一些特殊字符进行转换及在行尾加上
，这样可使输出的调试信息更便于阅读和分析。outdebug 方法代码如下：

```
    function outdebug($message){
            echo htmlspecialchars($message)."<br>\n";
    }
```

（6）使用 command 方法向 POP3 服务器发送命令。command 方法可以接收 3 个参数：参数说明如下：

- ☑ $command：发送给服务器的命令。
- ☑ $return_length：指定从服务器回送信息中截取多长作为命令回送标识。
- ☑ $return_code：指定标识的正确值。

如果服务器回送的第一个字符为“+”，则认为命令已正确执行；也可以用前面提到的“+OK”来判断。command 方法代码如下：

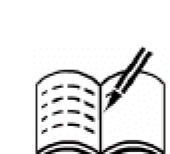
Note

```
<!--**************************mailclass.php******************************-->
	function command($command,$return_length=1,$return_code='+'){
		if($this->connection==0){//如果$this->connection 的值为 0，说明还没有连接服务器
			$this->err_str="没有连接上任何服务器，请检查网络连接!";
			return false;
		}
		if($this->debug)					//如果处于调试状态，则给出调试信息
		$this->outdebug(">>> $command");
		if(!fputs($this->connection,"$command\r\n")){//如果发送命令失败则给出信息
			$this->err_str="无法发送命令".$command;
			return false;
		 }else{
			$this->getresp();//如果发送命令成功，则用 getresp()方法来接收返回的信息
			if($this->debug)
			$this->outdebug($this->resp);
if(substr($this->resp,0,$return_length)!=$return_code){ //如果返回的信息不是以“+”开头，说明返回的信息无效
				$this->err_str=$command."命令，服务器返回无效：".$this->resp;
				return false;
			}else
				return true;
			}
		}
```

（7）使用 login()方法对登录到 POP3 服务器上的用户进行身份验证，如果验证通过则设置记录状态的数据成员$state 的值为 TRANSACTION。代码如下：

```
<!—**************************mailclass.php**************************-->
	function login($user,$password){
if($this->state!="AUTHORIZATION"){ //标志状态的值不为 AUTHORIZATION，说明还没有连接上服务器
			$this->err_str="还没有连接到服务器或状态不对!";
			return false;
		}
		if(!$this->apop){	//不需要以加密方式进行密码验证
		//向服务器发送 USER 命令，并通过返回信息的前 3 个字符是否为“+OK”来验证命令是否发送成功
			if(!$this->command("USER $user",3,"+OK"))
				return false;
		//向服务器发送 PASS 命令，并通过返回信息的前 3 个字符是否为“+OK”来验证命令是否发送成功
			if(!$this->command("PASS $password",3,"+OK"))
			    return false;
			}else{		//需要以加密方式进行密码验证
		//向服务器发送 APOP 命令，以加密的方式进行身份验证
				if(!$this->command("APOP  $user".md5($this->greeting.$password),3,"+OK"))
				return false;
			}
```





Note

```
        $this->state="TRANSACTION";   //如果验证通过则使记录状态的数据成员$state 的
值为 TRANSACTION
                return true;
            }
```

例如，下面是用户成功登录服务器时返回的信息。

```
+OK
+OK User successfully logged on
```

上面代码中第一个“+OK”说明 USER 命令发送成功，第二个“+OK”说明 PASS 命令发送成功。

（8）使用 stat()方法获取邮件总数量及总大小。stat()方法将向服务器发送 STAT 命令，从而获得邮件总数量和总大小。其代码如下：

```
<!--***************************mailclass.php******************************-->
    function stat(){
        if($this->state!="TRANSACTION"){          //判断用户是否已经成功登录
            $this->err_str="还没有连接到服务器或没有登录成功！";
            return false;
        }
        if(!$this->command("STAT",3,"+OK"))  //向服务器发送 STAT 命令
            return false;
        else{
            $this->resp=strtok($this->resp," ");        //取得“+OK”
            $this->messages=strtok(" ");                //取得邮件总数
            $this->size=strtok(" ");                    //取得邮件总字节数
            return true;
        }
    }
```

例如，下面是 STAT 命令发送成功后返回的信息。

```
+OK 3 1504
```

上面代码中，“+OK”说明命令发送成功；3 表示邮箱中有 3 个邮件；1504 表示邮件总大小为 1504 字节。

在 stat()方法中应用 strtok()函数对字符串进行分割。strtok()函数的语法格式如下：

```
string strtok(string str1, string str2)
```

上述代码实现了从 str1 中返回 str2 首字母前的字符串。如果某程序中有多个 strtok()函数，只需要第一个 strtok()函数给出 str1 的值，第二个以后 str1 的默认值为上次返回值后剩余的字符串。

（9）定义 listmail()方法获取指定邮件的大小。使用 listmail()方法获取指定邮件大小的代码如下：

```
<!—*************************mailclass.php**************************-->
    function listmail($mess=NULL,$uni_id=NULL){
        if($this->state!="TRANSACTION"){   //判断用户是否已经成功登录
```




```
                $this->err_str="还没有连接到服务器或没有登录成功!";
                return false;
            }
            if($uni_id)                 //如果指定$uni_id，则使用 UIDL 命令
                $command="UIDL";
            else
                $command="LIST";   //如果没指定$uni_id，则使用 LIST 命令
            if($mess)
                $command.=$mess;   //如果指定$mess，则使用$mess 所指定的命令
            if(!$this->command($command,3,"+OK")){  //利用 command()方法发送命令
                return false;
            }else{
               $i=0;
                $this->mail_list=array();   //定义二维数组，用来保存每个邮件的大小
                $this->getresp();           //获取服务器返回的信息
                while($this->resp!="."){    //.为邮件结束的标志
                    $i++;
                    if($this->debug){
                    $this->outdebug($this->resp);
                    }
                    if($uni_id){
                        $this->mail_list[$i]['num']=strtok($this->resp," ");
                        $this->mail_list[$i]['size']=strtok(" ");   //获取某个邮件的大小
                    }else{
                        $this->mail_list[$i]["num"]=intval(strtok($this->resp," "));
                        $this->mail_list[$i]["size"]=intval(strtok(" "));
                    }
                    $this->getresp();
                }
                return true;
            }
        }
```

（10）定义 getmail()方法获取邮件头及邮件内容。使用 getmail()方法获取邮件头及邮件内容的代码如下：

```
        function getmail($num=1,$line=-1){
            if($this->state!="TRANSACTION"){          //判断用户是否已经成功登录
                $this->err_str="不能收邮件，还没有连接服务器或登录成功！";
               return false;
            }
           if($line<0)         //如果$line<0 则返回所有的邮件内容
                $command="RETR $num";
            else               //如果$line>=0 则返回邮件前$num 行内容
                $command="TOP $num $line";
            if(!$this->command("$command",3,"+OK"))  //向服务器发送命令
                return false;
            else{
                $this->getresp();            //如果命令发送成功，则获取返回信息
                $is_head=true;
```



Note

```
            while($this->resp!="."){
                if($this->debug)
                $this->outdebug($this->resp);
                if(substr($this->resp,0,1)==".")    //.号为邮件结束的标志
                $this->resp=substr($this->resp,1,strlen($this->resp)-1);
                if(trim($this->resp)=="")    //邮件头与邮件内容之间以空行分隔
                    $is_head=false;
                if($is_head)
                    $this->head[]=$this->resp;  //将邮件头每一行的内容放到 head[]数组中
                else
                $this->body[]=$this->resp;  //将邮件内容的每一行放到body[]数组中
                    $this->getresp();
            }
            return true;
        }
    }
```

（11）定义 dele()方法删除指定邮件。使用 dele()方法删除指定邮件的代码如下：

```
    function dele($num){
        if($this->state!="TRANSACTION"){                //判断用户是否已经成功登录
            $this->err_str="不能删除邮件";
          return false;
        }
        if(!$num){
            $this->err_str="删除的邮件参数不对！ ";
            return false;
        }
        if($this->command("DELE $num",3,"+OK"))        //删除$num 指定的邮件
            return true;
        else
            return false;
    }
```

（12）定义 close()方法断开与服务器的连接，代码如下。

```
    function close(){                              //dele 方法的作用是删除$num 指定的邮件
        if($this->connection!=0){
            if($this->state=="TRANSACTION")
$this->command("QUIT",3,"+OK"); //向服务器发送 QUIT 命令，从而断开与服务器的连接和执行 DELE 命令
                fclose($this->connection);
                $this->connection=NULL;
                $this->state="DISCONNECTED";
            }
        }
    }
```



FTP 管理系统

（自学视频、源程序：配套资源\mr\14\）

FTP（File Transfer Protocol）是 Internet 上文件传输通用协议，位于 TCP/IP 协议的应用层。利用该协议，用户可以将一个完整的文件从一台计算机复制到另一台计算机上。不过，在使用 FTP 传输文件之前，需要先登录 FTP 服务器（可以通过已设置的 FTP 用户名和密码登录服务器）。本章将向读者展现一个多功能 FTP 管理系统，主要实现本地系统与远程 FTP 服务器之间文件的上传、下载和管理文件等功能。通过本章的学习，读者能够学到以下内容。

- 遍历 FTP 服务器目录
- 获取本地文件的图标
- 实现 FTP 文件上传、下载
- 本地文件与 FTP 文件的维护操作





14.1　FTP 管理系统概述

14.1.1　模块概述

FTP 管理系统是一个可以执行常用操作的上传、下载工具，通过它用户可以方便、快捷地访问 FTP 服务器，彼此独立地传输文件，即多个用户在分别执行各自的上传、下载和维护等操作时互不影响。另外，FTP 管理系统具有良好的远程管理性，用户可以通过互联网登录 FTP 服务器，执行远程管理操作。

14.1.2　功能结构

FTP 管理系统包括由根目录操作和子目两大基本模块构成。其中根目录操作包括查看文件、查看目录、删除目录、创建目录和上传文件等；子目录操作包括查看文件、查看目录、更改文件、移动文件、删除文件、创建目录和上传文件等。FTP 管理系统的功能结构如图 14.1 所示。

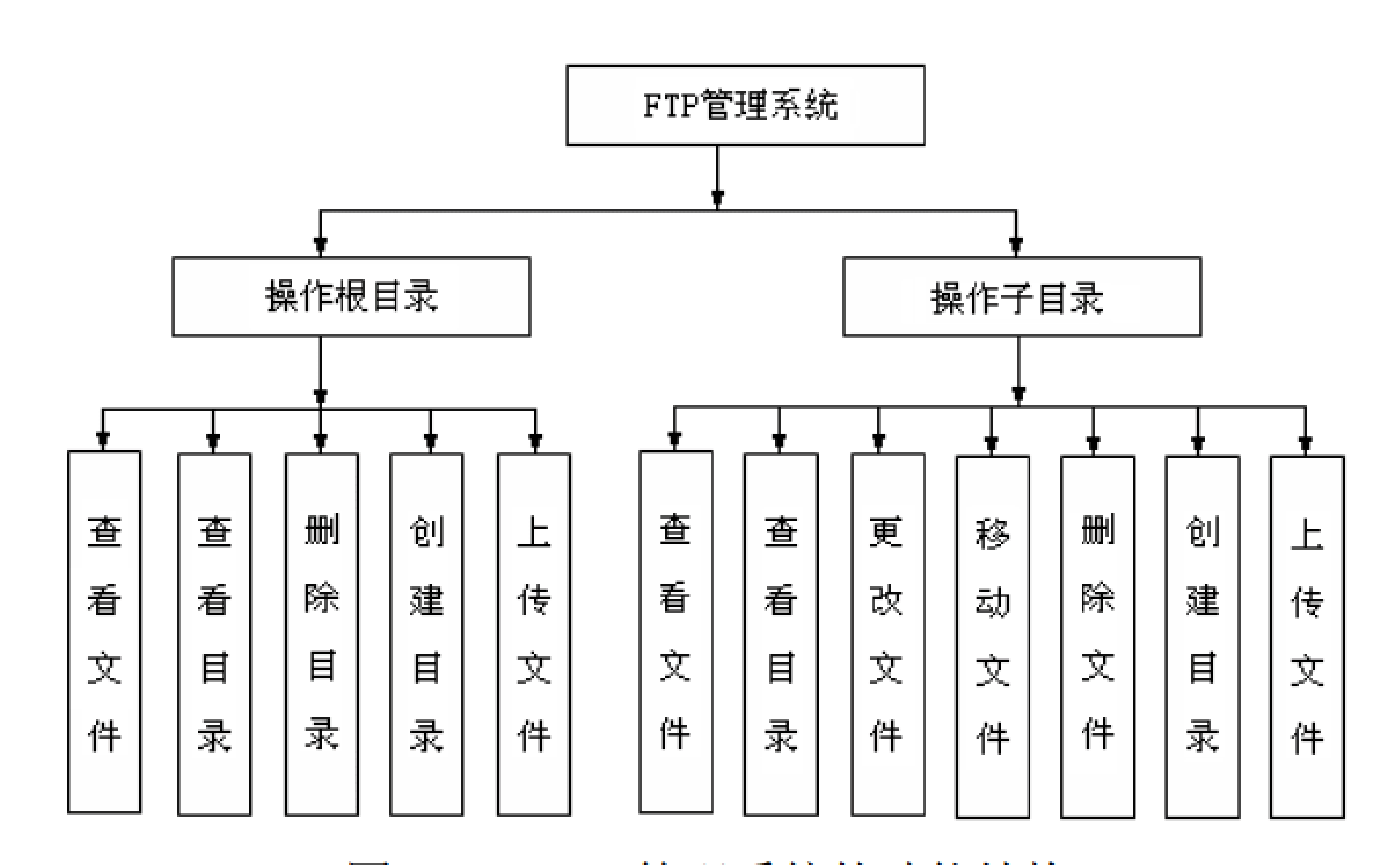
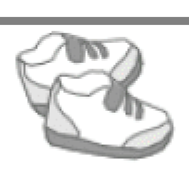


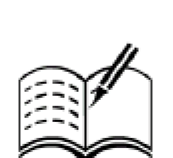
图 14.1　FTP 管理系统的功能结构

14.1.3　程序预览

为了让读者对本系统有个初步的了解和认识，下面给出本系统的几个页面运行效果图，如果想查看完整的效果图，参见配书光盘源程序。

FTP 管理系统的登录页面和主页的运行效果分别如图 14.2 和图 14.3 所示。





Note



图 14.2　登录页面



图 14.3　FTP 管理系统的主页

子目录下的文件详细信息运行效果如图 14.4 所示。



图 14.4　子目录下文件详细信息

单击文件名称超链接，下载 FTP 文件的运行效果如图 14.5 所示。单击“移动”超链接，移动文件到指定文件目录的运行效果如图 14.6 所示。

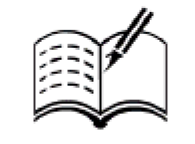


图 14.5　FTP 文件下载运行效果



图 14.6　FTP 文件移动运行效果





14.2 FTP 环境搭建

14.2.1 FTP 服务器端概述

FTP 服务器端是一种可以将计算机或服务机设置成 FTP 服务器的软件工具。通过 FTP 软件，用户或其他使用者就能够使用 FTP 协议，通过互联网或者同一网络上的任何一台计算机与 FTP 服务器连接，执行文件或目录的复制、移动、创建和删除等操作。这里提到的 FTP 协议是专门被用来规定计算机之间进行文件传输的标准和规则。正是因为有了像 FTP 这样的专门协议，才使得人们能够通过不同类型的计算机，使用不同类型的操作系统，实现不同类型文件的相互传递。

目前 FTP 软件的种类繁多，其功能各有千秋。Serv-U 是其中的佼佼者，应用最为广泛。Serv-U 软件可以提供以下功能：

- ☑ 符合 Windows 标准的用户界面友好亲切，易于掌握。
- ☑ 通过限制同一时间最大的用户访问人数确保 PC 的正常运转。
- ☑ 安全性能出众，在目录和文件层次都可以设置安全防范措施。
- ☑ 能够为不同用户提供不同设置，支持分组管理数量众多的用户。
- ☑ 可以基于 IP 对用户授予或拒绝访问权限。
- ☑ 支持文件上传和下载过程中的断点续传。
- ☑ 支持拥有多个 IP 地址的多宿主站点。
- ☑ 能够设置上传和下载的比率、硬盘空间配额、网络使用带宽等，从而能够保证用户有限的资源不被大量的 FTP 访问用户所消耗。
- ☑ 可作为系统服务后台运行。
- ☑ 可自由设置在用户登录或退出时的显示信息，支持具有 UNIX 风格的外部链接。

14.2.2 Serv-U 的安装和配置

为了保证 FTP 管理系统能够正常运行并发挥其应有的功效，在使用该系统前应在远程服务器上搭建 FTP 服务环境。本系统选择 Serv-U 来搭建 FTP 服务环境。下面介绍 FTP 服务环境的架设过程。

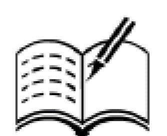
（1）通过互联网下载 Serv-U FTP Server 安装包，然后将其解压缩到指定文件目录下，双击 ServUSetup.exe 安装文件。

（2）在弹出界面中选择语言为中文（简体），单击“确定”按钮，进入“安装向导”界面，如图 14.7 所示。

（3）单击“下一步”按钮，进入许可协议查看界面。选择“我接受协议”单击“下一步”按钮，进入“选择目标位置”界面，如图 14.8 所示。在该界面中设置 Serv_U 安装路径，然后单击“下一步”按钮。

（4）进入“准备安装”界面，确认无误后单击“安装”按钮，开始安装 Serv_U，如图 14.9 所示。





Note



图 14.7 “安装向导”界面

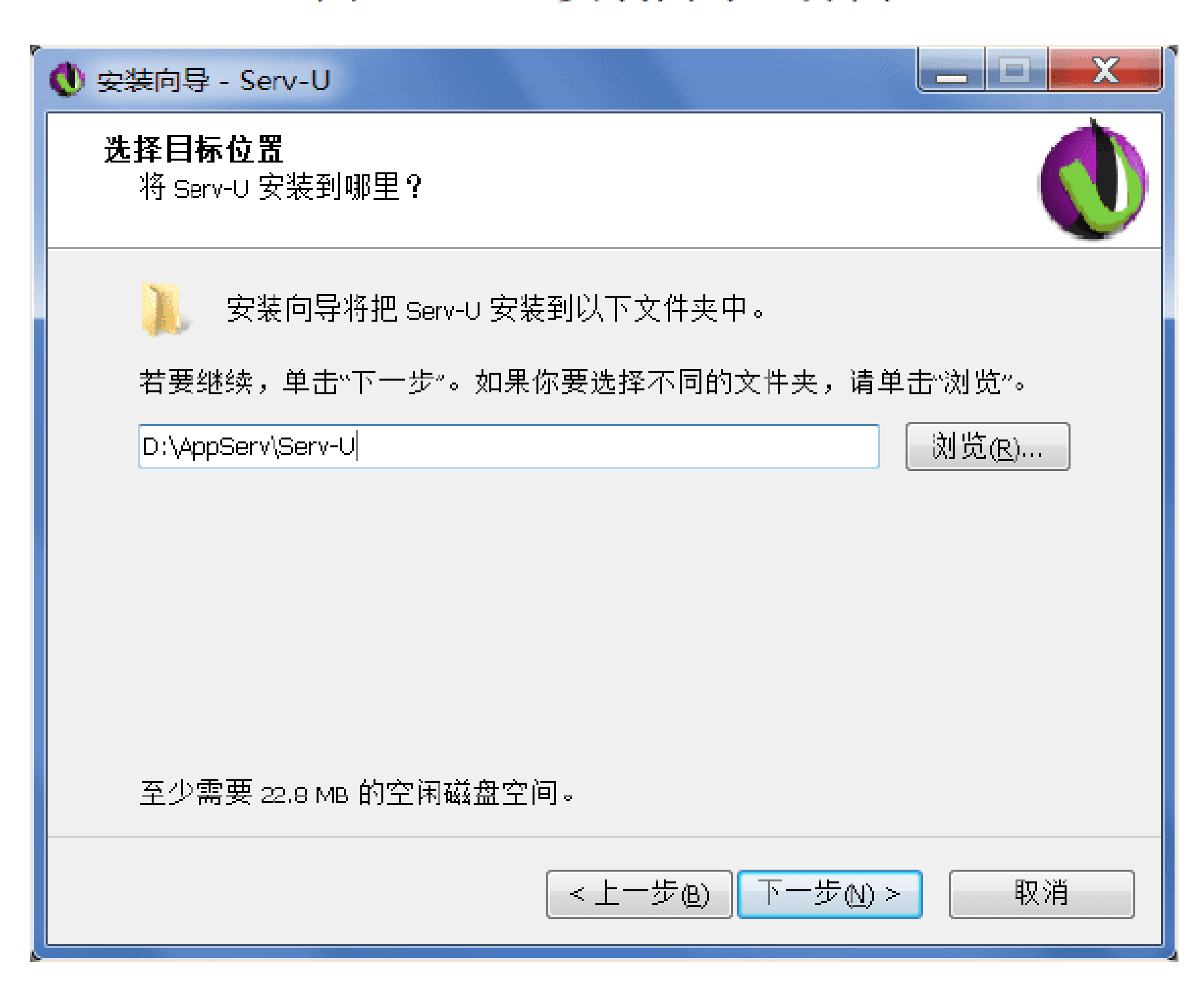


图 14.8 “选择目标位置”界面

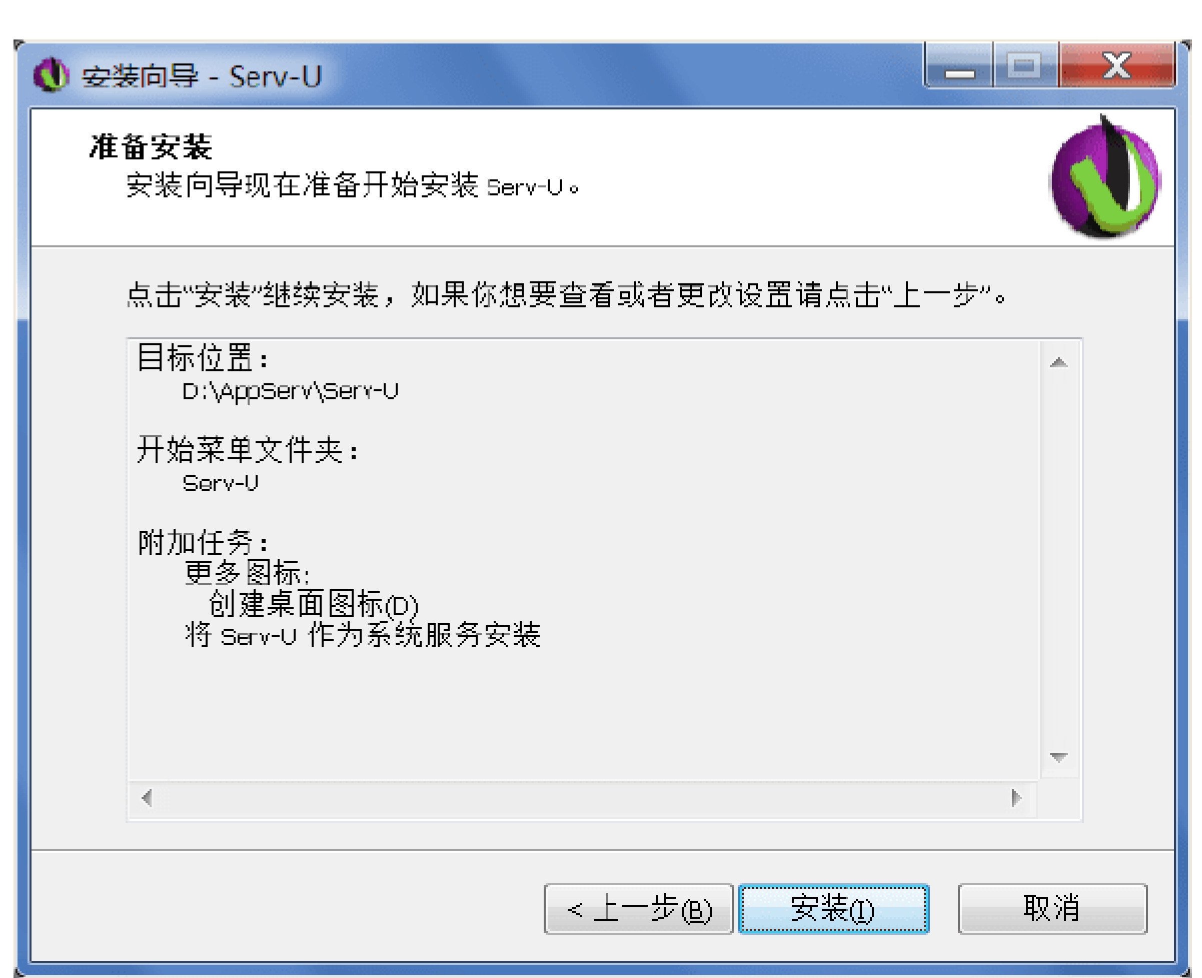


图 14.9 准备安装界面





（5）当 Serv_U 的所有部分安装完毕后，将弹出如图 14.10 所示的“完成 Serv_U 安装”界面。



图 14.10　完成 Serv_U 安装

（6）单击“完成”按钮，这样 FTP 服务器端软件 Serv-U 安装完毕。

此时运行 Serv-U 软件，系统会提示用户设置相应的必要参数。下面讲解如何配置 Serv-U。

（1）定位到新添加的文件目录，找到可执行文件 Serv-U-Tray.exe，双击该文件即可启动 Serv_U。进入控制管理台（如图 14.11 所示）后，首先需要设置上传。



图 14.11　管理控制台

（2）单击“域详细信息”，进入“域”界面。如果是首次设置域信息，则需要单击“新建域”按钮新建域，如图 14.12 所示。





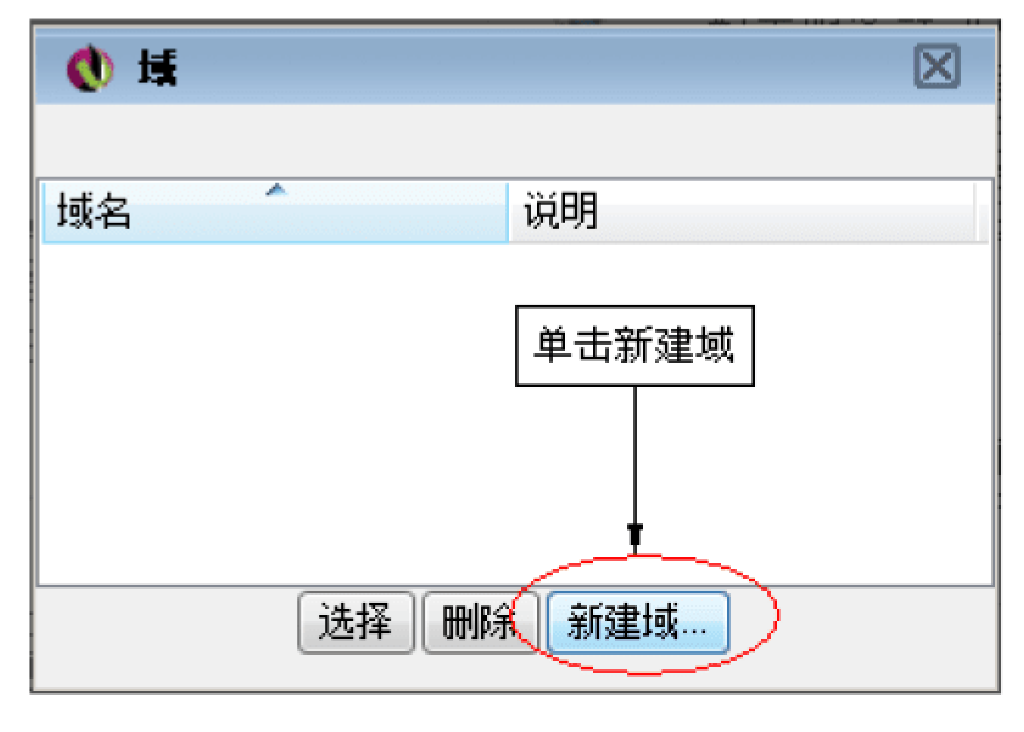


图 14.12　单击“新建域”按钮设置域信息

（3）进入“域向导”界面，需要进行 4 步设置，即设置域名称；设置 FTP 协议、端口；设置可用 IP 地址；设置密码加密方式。其中设置 FTP 端口、密码加密方式的界面如图 14.13 和图 14.14 所示。

注意：

如果只是在本机上测试，建议不要更改参数设置。

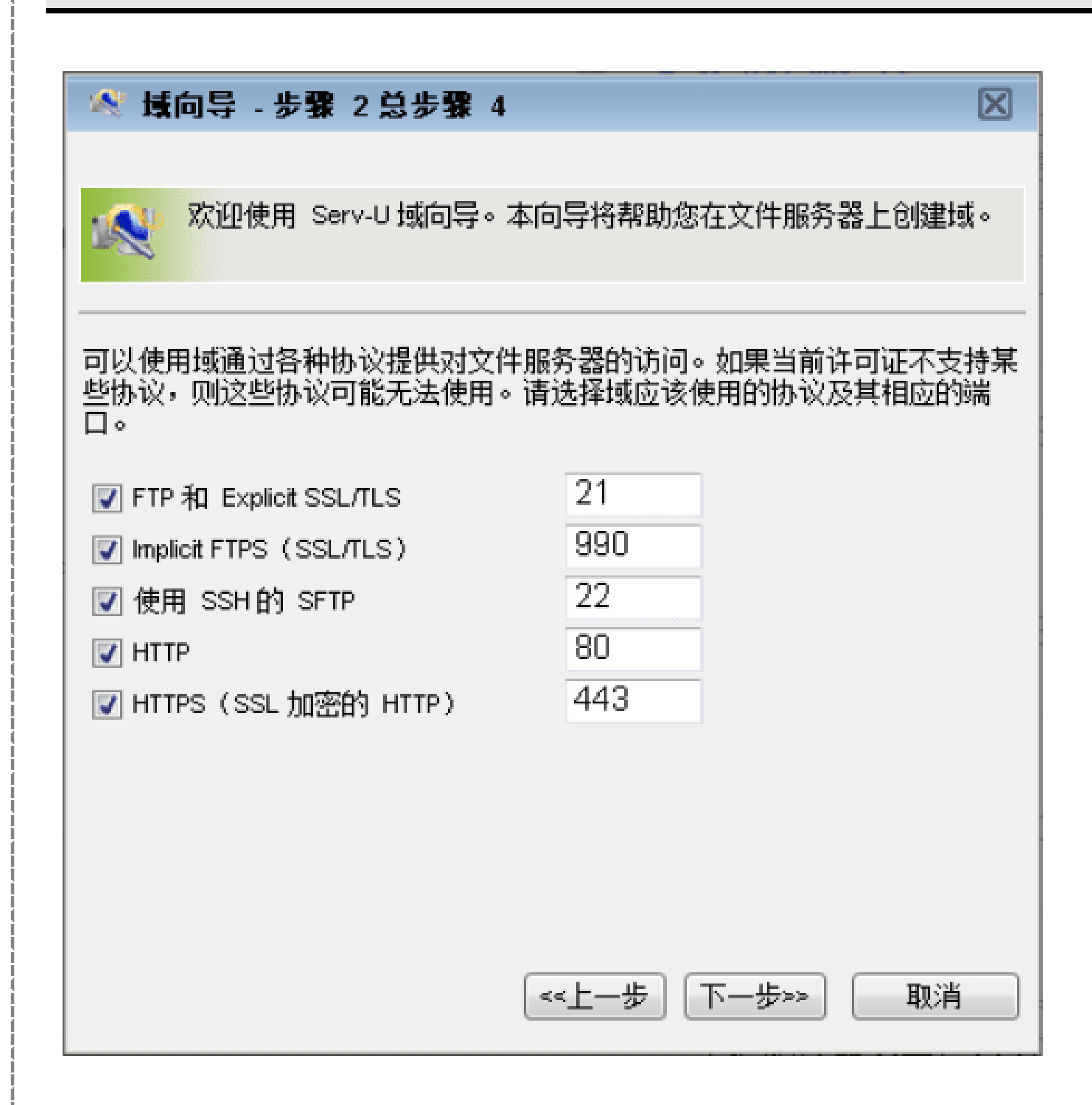
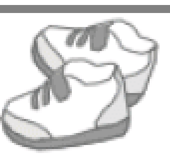


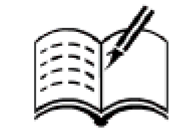
图 14.13　FTP 端口设置

图 14.14　FTP 密码加密方式设置

（4）填写域名称，然后单击“下一步”按钮，直至设置完成。

（5）完成新建域后，返回管理控制台。单击“添加新用户”选项创建新用户。单击“创建、修改和删除用户账户”，进入“用户属性”界面，如图 14.15 所示。选择“用户信息”选项卡，在其中依次设置登录 ID、全名、密码、根目录、管理权限、SSH 公共密钥路径、账户类型、说明等选项，然后单击“保存”按钮，即可完成用户的添加。





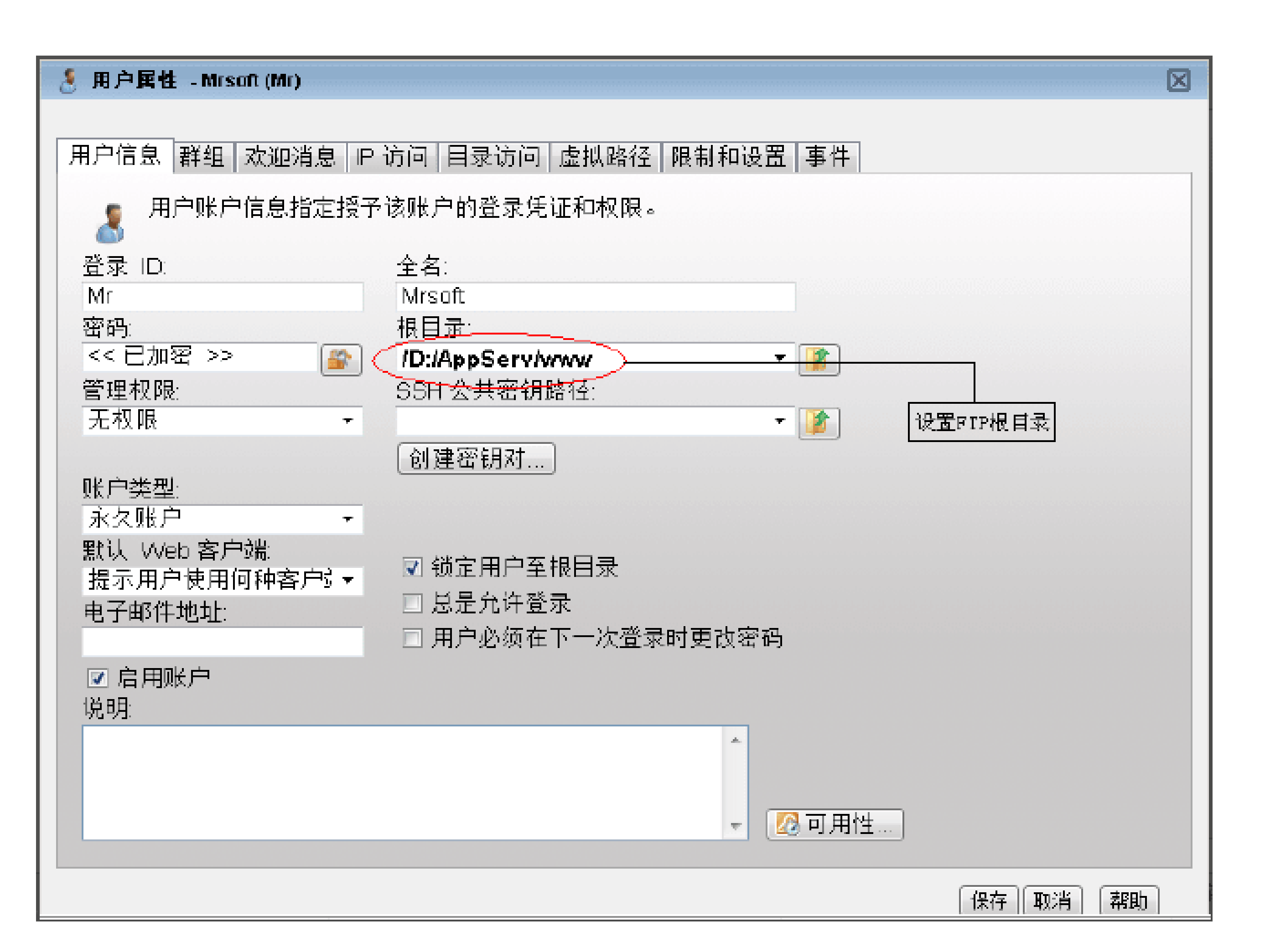



图 14.15 “用户属性”界面

说明：

这里建议读者将根目录设置为运行 PHP 环境的目录，以便在执行远程操作时更方便地访问 PHP 服务器目录，使 FTP 管理系统发挥更大的作用。笔者在设置 Serv-U 服务器端用户时，将 FTP 访问的根目录设置成 Appserv 集成环境所在的目录。

至此，FTP 服务器端的架设和参数设置介绍完毕。如需要对其他参数进行设置，建议读者参考官方提供的有关资料。由于篇幅限制，其他的设置参数不能一一列举。

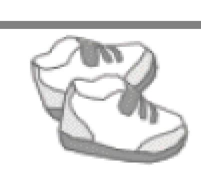
14.3 用户登录

14.3.1 用户登录功能概述

这里的用户登录功能是指 FTP 服务器端对用户输入的 IP 地址、用户名等 FTP 连接信息进行验证。与普通登录验证方式不同的是，用户在登录页面提交的验证信息需要与指定服务器的预设登录信息进行对比，如果用户输入的连接信息与预设信息匹配，则用户可以顺利连接 FTP 服务器，否则将不能连接 FTP 服务器。这样就可以使用户安全地连接 FTP 服务器。登录页面的运行效果如图 14.16 所示。

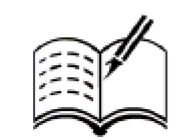


图 14.16 登录页面的运行效果



用户登录功能的实现涉及两个核心文件：index.php（登录表单页）、config.php（系统连接公共文件）。

14.3.2 登录页面设计


Note

登录表单页（index.php）的结构相对比较简单，主要包含连接 FTP 服务器的“IP 地址”文本框、“用户名”文本框、“密码”文本框和“登录”按钮。登录页面中使用的元素如表 14.1 所示。

表 14.1 登录页面主要元素

元 素 名 称	元素属性值	说　明
script	"js/check.php"	登录信息是否为空验证脚本
text	id="ip"name="ip"	连接服务器 IP 地址
text	id="username"name="username"	连接用户名
password	id="password"name="password"	连接用户密码
submit	name="login" value="登录"	“登录”按钮

在登录表单页中，通过 check.js 脚本中的 chklogin()函数对提交的登录信息进行验证。在 chklogin()函数中，通过获取 ID 值为 login 的表单中的元素来判断文本框是否为空。如果表单元素中文本框为空，则将焦点定位到该元素上。其代码如下：

```
function chklogin(){
        if(login.ip.value==""){
            alert("请填写连接 IP");login.ip.focus();return false; //判断连接 IP 文本框是否为空
        }
        if(login.username.value==""){
            alert("请填写连接用户名");login.username.focus();return false;    //判断连接用户名是否为空
        }
        if(login.password.value==""){
            alert("请填写连接密码");login.password.focus();return false;     //判断连接密码是否为空
        }
        login.submit();                                          //提交 login 表单
}
```

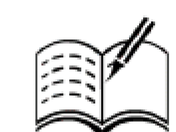
14.3.3 系统连接公共文件

检测 FTP 登录用户信息的操作在 config.php 文件中完成。首先获取表单提交的登录数据，将其赋给指定的 SESSION 变量。其代码如下：

```
<?php
if(isset($_POST['login'])){                         //判断是否以 POST 方式获取 login 传递参数
        $_SESSION['ip'] = $_POST['ip'];             //获取验证 IP 地址并赋给 SESSION 变量
        $_SESSION['user'] = $_POST['username'];     //获取验证用户名并赋给 SESSION 变量
        $_SESSION['pwd'] = $_POST['password'];      //获取验证密码并赋给 SESSION 变量
```

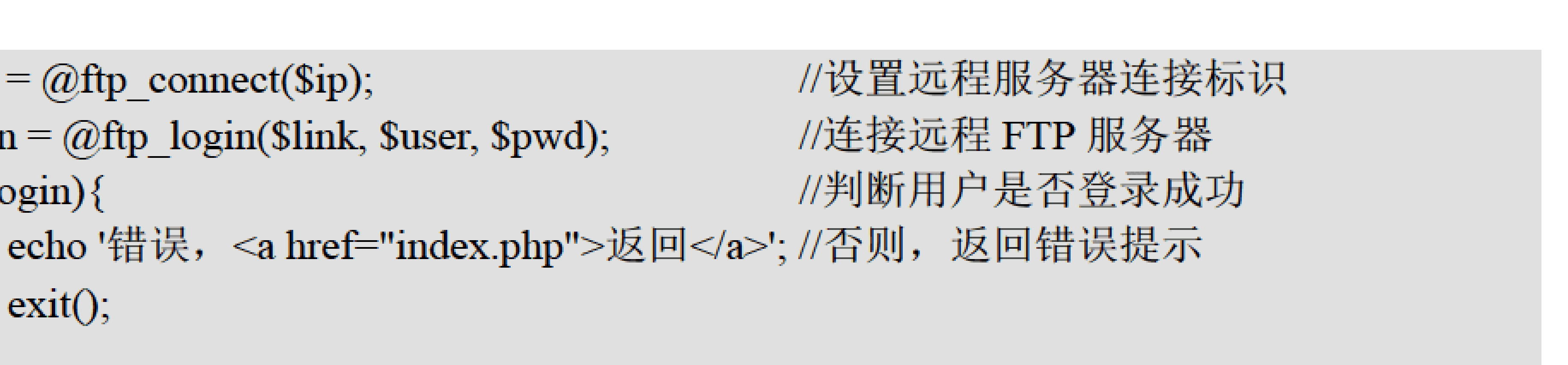


```
}
$ip = $_SESSION['ip'];                          //传递 SESSION 变量给相应变量
$user = $_SESSION['user'];
$pwd = $_SESSION['pwd'];
```



Note

然后，将获取的 IP 地址、用户名、密码作为连接参数，远程连接 IP 地址指向的服务器。如果连接成功则显示 main.php 页面的其他内容；如果失败则返回错误信息。相关代码如下：



```
$link = @ftp_connect($ip);                      //设置远程服务器连接标识
$login = @ftp_login($link, $user, $pwd);        //连接远程 FTP 服务器
if(!$login){                                    //判断用户是否登录成功
    echo '错误，<a href="index.php">返回</a>'; //否则，返回错误提示
    exit();
}
?>
```

说明：

本系统将 config\config.php 文件作为 FTP 连接的公共文件。为了避免重复书写 FTP 连接过程，在实现其他 FTP 功能时，只需通过 include 语句调用该公共文件即可完成与 FTP 服务器的连接。

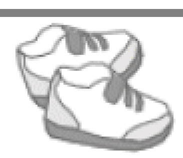
14.4　根目录操作

14.4.1　根目录功能概述

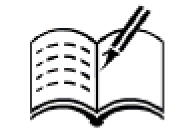
用户登录 FTP 管理系统后，可以遍历用户在 FTP 服务器中指向的系统根目录。用户可以单击根目录进入次级根目录，并可对根目录执行操作。登录 FTP 管理系统的运行效果如图 14.17 所示。



图 14.17　FTP 管理系统运行效果



其中根目录的主要功能包括：查看根目录名称、删除目录、创建文件目录、上传文件到根目录等。其主要是在 main.php（系统管理页）、mkdir.php（创建文件目录页）、rmkdir.php（删除文件处理页）等文件中实现的。



14.4.2 遍历文件根目录

遍历文件根目录，就是进入系统管理页（main.php），应用 ftp_rawlist()函数结合 foreach 语句输出根目录中的所有文件目录名称。单击文件目录名称对应的超链接可进入次级目录；如果想返回根目录或上级目录，单击“返回上级目录”即可。

1. 遍历根目录名称

应用 ftp_pwd()、ftp_rawlist()函数获取路径名称、目录列表以及目录下的文件详细信息，代码如下。

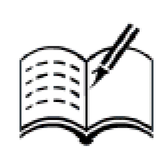
```
<?php
$dirname = ftp_pwd($link);                          //返回当前目录名
if($dirname != '/'){                                //判断当前目录中是否存在'/'
    $dirname .= '/';                                //不存在则将路径名称赋值'/'
}
$flist = ftp_rawlist($link, $dirname);              //返回当前目录下详细列表
$sdir = array();                                    //更改变量$sdir 数据类型
foreach($flist as $value){                          //循环输出目录详细列表
    $array = explode(' ', rtrim($value));       //将去右空白的元素以空格作为分割符分隔
    $alen = count($array);                          //获取数组元素个数
    $p = substr($array[0], 0, 1);                   //截取返回数组长度
    if($p == 'd'){
        if($array[$alen-1] != '.' && $array[$alen-1] != '..'){
            $sdir[] = $array[$alen-1];
        }
    }
}
?>
```

将获取的当前目录名称、目录列表以及文件详细列表，在显示页（main.php）中应用 foreach 语句循环输出。其代码如下：

```
<div class="mainleft">
   <table border="0" cellpadding="0" cellspacing="0">
     <?php
        //以下循环输出根目录中的目录名称列表
        foreach($sdir as $value)
           echo'<tr>
                 <td height="10" width="30">  </td>
                 <td height="25" width="75" align="center" class="STYLE1">
<a href="main.php?dir='.$dirname.$value.'">'.substr($value,0,12).'</a>
</td>
                 <td width=50><a href="rmkdir.php?dir='.$dirname.$value.'">删除</a></td>
```



```
                  <td height="5" width="30">  </td>
                </tr>';
        ?>
    </table>
</div>
```



Note

2. 返回上级目录

返回上级目录是通过单击“返回上级目录”超链接来实现的。其实现原理是，根据 URL 地址中传递的目录路径 dir，应用 ftp_chdir 改变当前路径为指定路径。相关代码如下：

```
<?php
if(isset($_GET['dir'])){                             //获得以 GET 方式传递的 URL 参数
    ftp_chdir($link, $_GET['dir']);                  //改变当前路径为指定路径
}
if(isset($_GET['up'])){                              //判断用户是否单击"返回上级目录"
    ftp_cdup($link);                                 //改变上级目录为指定路径
}
if(isset($_GET['down'])){                            //判断用户是否单击文件目录名称
    ftp_get($link, $_GET['down'], $_GET['down'], FTP_BINARY);//进入下级文件目录
}
?>
```

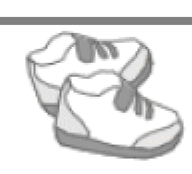
14.4.3 删除文件目录

在系统管理页中，每一个文件目录名称后面都有一个“删除”超链接。单击“删除”超链接，将文件目录名称作为参数传递到 rmkdir.php 文件，完成文件目录的删除操作。相关代码如下：

```
<?php
session_start();                                 //开启全局会话
$ip = $_SESSION['ip'];                           //将 SESSION 变量中的数据赋值给对应变量
$user = $_SESSION['user'];
$pwd = $_SESSION['pwd'];
$link = @ftp_connect($ip);                       //连接到指定 FTP 地址
$login = @ftp_login($link, $user, $pwd);         //连接服务器
$file = $_GET['dir'].$_GET['file'];              //设置删除文件位置
if(ftp_delete($link, $file)){                    //判断文件是否被删除
    echo '<script>alert("删除文件成功！");history.back();</script>';     //删除成功提示
}else{
    echo '<script>alert("删除文件失败！");history.back();</script>';     //删除失败提示
}
ftp_close($link);                                //关闭 FTP 连接
?>
```

14.4.4 在根目录中创建新目录

单击系统管理页中的“创建目录”按钮，跳转到 mkdir.php 页面，填写要在根目录中



Note

创建的新目录名称，然后将数据提交到 mkdir_chk.php 文件中，即可完成新目录的创建。

在 mkdir.php 文件中，完成目录名称的添加。其代码如下：

```
<link href="css/mkdir.css"  type="text/css" rel="stylesheet" />
<script src="js/check.js"></script>
<form id="mkdir" action="mkdir_chk.php?dir=" method="post" onSubmit="return chkname();">
<input id="dirname" class="inpt1" type="text" name="filename" />
<input  type="submit" value="提交" />
<input  type="reset" value="重置" />
</form>
```

在 mkdir_chk.php 文件中，获取表单提交的目录名称，应用 ftp_mkdir()函数在 FTP 服务器根目录中创建新文件目录。其代码如下：

```
<?php
session_start();
$ip = $_SESSION['ip'];                          //将 SESSION 变量中的数据赋值给对应变量
$user = $_SESSION['user'];
$pwd = $_SESSION['pwd'];
if(!empty($_GET["dir"])){                       //判断是否通过 GET 方式获取文件目录路径
    $path=$_GET["dir"];                         //将路径名称赋值给对应变量
}else{
    $path="";                                   //否则为根目录
}
$dirname=$path.$_POST["filename"];              //指定创建文件路径
$link = @ftp_connect($ip);                      //连接指定 FTP 地址
$login = @ftp_login($link, $user, $pwd);        //连接 FTP 服务器
$result = ftp_mkdir($link, $dirname);           //创建文件目录
if($result) {                                   //判断文件目录是否被创建
    echo '<script>alert("目录'.$dirname.'\n 创建成功！");window.close();;</script>';  //提示创建成功
}else {
    echo '<script>alert("目录'.$dirname.'\n 创建成功！");history.back();</script>';  //提示创建失败
}
ftp_close($link);                               //关闭 FTP 连接*/
?>
```

14.4.5　上传文件到根目录下

在系统管理页中，单击“上传文件”按钮，跳转到 upfile.php 页面中，选择要上传的本地文件，然后将数据提交到 upfile_chk.php 文件中，即可实现将本地文件上传到服务器指定的文件夹下。

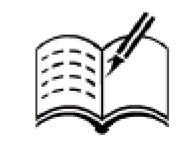
在 upfile.php 文件中，创建表单，添加文件域，将本地文件提交到 upfilechk.php 文件中。其代码如下：

```
<form action="upfile_chk.php" method="post" enctype="multipart/form-data">
  <input class="inpt1"  type="file" name="file" />
```



```
    <input class="inpt1" type="submit" value="提交" />
</form>
```

在 upfile_chk.php 文件中，获取表单中提交的本地文件信息，应用 ftp_put()函数将文件上传到服务器的根目录中。相关代码如下：



Note

```
<?php
session_start();                                          //开启全局会话
include_once("config/config.php");                        //加载公共连接文件
$up=$_FILES["file"]["tmp_name"];                          //获取上传文件
$file=$_FILES["file"]["name"];                            //定义上传文件名称
if(ftp_put($link,$file,$up,FTP_BINARY)){                  //上传文件到服务器根目录中
echo '<script>alert("上传'.$file.'\n 成功！ ");window.close();</script>';     //提示上传成功
}else {
echo '<script>alert("上传'.$file.'\n 失败！ ");history.back();</script>';     //提示上传失败
}
ftp_close($link);                                         //关闭 FTP 连接
?>
```

14.5 子目录操作

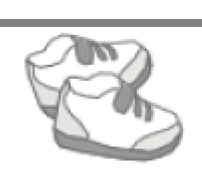
14.5.1 子目录功能概述

单击任意根目录，即可进入次级目录，并在右侧窗口中显示子目录中包含的所有文件。子目录中部分操作与根目录操作大同小异，但是进入子目录后，不但可以操作子目录，还可以对子目录下的文件执行多种操作。进入子目录的效果如图 14.18 所示。

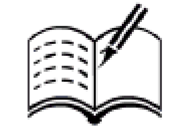


图 14.18 子目录运行效果

用户对子目录执行的操作包括：查看子目录文件、删除子目录、在子目录中创建新文件目录、上传文件到当前子目录、更改子目录下的文件名称、移动文件到其他目录中，以



及下载远程文件到本地计算机中。对于目录的操作，主要是在 main.php（系统管理页）、removefile.php.php（删除文件处理页）、movefile.php（移动文件处理页）等文件中完成的。



Note

说明：

由于目录的创建和上传文件等功能与根目录操作大体相同，在此不再赘述。下面只对子目录中特有的一些功能进行讲解。

14.5.2 浏览子目录下的文件

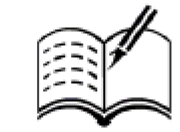
登录 FTP 管理系统后，单击任意根目录即可进入次级目录，并在右侧窗口中显示子目录中包含的所有文件。从中可以查看文件名称、文件大小、最后修改时间等，还可以对该目录下的文件执行更改名称、删除和移动等操作。在 main.php 文件中，遍历子目录中所有文件相关信息的关键代码如下。

```
<?php
$dirname = ftp_pwd($link)                          //返回当前目录名
if($dirname != '/'){                               //判断当前目录中是否存在'/'
        $dirname .= '/';                           //不存在则将路径名称赋值'/'
}
$sfile = ftp_nlist($link, $dirname);               //返回当前根目录列表
if(!$sfile){                                       //判断根目录下是否为空
        $sfile = array();                          //更改变量$sfile 数据类型
}
$flist = ftp_rawlist($link, $dirname);             //返回当前目录下详细列表
?>
```

通过 foreach 语句循环输出子目录中的详细内容。其关键代码如下：

```
<table border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#BAE8BC">
      <tr>
          <td height="35"    colspan="6" bgcolor="#FFFFFF"><div style=" margin:auto 20px;
text-align:right;" >
              <input type="button" value="" class="btn3" onclick="javascript:Wopen=open
('path_mkdir.php?
dir=<?php echo $dirname; ?>','','height=220,width=350,scrollbars=no');">
              <input type="button" class="btn4" onclick="javascript:Wopen=open('path_upload.php?
dir=<?php echo $dirname; ?>','','height=220,width=350,scrollbars=no');"></div>
</td>
      </tr>
      <tr>
          <td height="28" align="center" bgcolor="#FFFFFF">文件名称</td>
          <td align="center" bgcolor="#FFFFFF">文件大小</td>
          <td align="center" bgcolor="#FFFFFF">最后修改时间</td>
          <td align="center" bgcolor="#FFFFFF">更改名称</td>
          <td align="center" bgcolor="#FFFFFF">删除文件</td>
          <td align="center" bgcolor="#FFFFFF">移动文件</td>
      </tr>
```





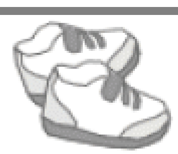

```
 <?php
     foreach($sfile as $value){                              //循环输出文件列表
     ?>
    <tr>
        <td width="250" height="30" align="center" bgcolor="#FFFFFF">
<a href="#" onclick="javascript:Wopen=open('download.php?dir=<?php echo $dirname; ?>
&filename=<?php echo $value;?>','','height=220,width=350,scrollbars=no');"><?php echo $value; ?></a>
</td>
<td width="70" height="30" align="center" bgcolor="#FFFFFF"><?php echo round(ftp_size($link,$value)/1024,2);?> K
</td>
        <td width="200" height="30" align="center" bgcolor="#FFFFFF">
<?php echo date('Y-m-d H:i:s',ftp_mdtm($link,$value)); ?>
</td>
     <td width="60" height="30" align="center" bgcolor="#FFFFFF">
<a href="#" onclick="javascript:Wopen=open('edit.php?file=<?php echo $value; ?>
&dir=<?php echo $dirname; ?>','','height=220,width=350,scrollbars=no'); ">更改</a>
</td>
        <td width="60" height="30" align="center" bgcolor="#FFFFFF">
<a href="removefile.php?file=<?php echo $value; ?>&dir=<?php echo $dirname;?>">删除</a>
</td>
        <td width="60" height="30" align="center" bgcolor="#FFFFFF">
<a href="#" onclick="javascript:Wopen=open('movefile.php?dir=<?php echo $dirname; ?>
&filename=<?php echo $value;?>','','height=220,width=350,scrollbars=no');">移动</a></td>
     </tr>
    <?php
           }                                      //foreach 语句结束
    ?>
</table>
```

14.5.3 远程文件下载

作为一款 FTP 管理工具，下载远程服务器中文件这一功能必不可少。与普通 Web 程序下载功能不同，FTP 服务器上的文件并不能直接通过单击超链接形式下载，因为 FTP 服务器指定的目录不一定是 Web 服务器或提供服务的环境目录。在此需要通过 ftp_get()函数来执行远程文件下载操作。

单击文件名超链接，进入 download.php 文件下载页面，填写正确的本地下载路径，然后单击“下载”按钮，将数据提交到 download.php 文件中，即可将远程服务器中的文件下载到本地。download.php 文件的代码如下：

```
<?php
session_start();                              //开启全局会话
include_once("config/config.php");            //加载公共连接文件
if(!empty($_POST["savepath"])){               //判断是否存在下载路径
$absopath=$_POST["savepath"];                 //获取下载路径
$dir=$_GET["dir"];                            //获得下载的文件路径
```




```
$dwname=$_GET["filename"];                          //获取下载文件名称
$file=$dir.$dwname;                                 //定义下载文件
$down=$absopath.$dwname;                            //定义下载路径
if(ftp_get($link,$down,$file,FTP_BINARY)) {         //判断是否从 FTP 服务器中下载文件
    echo '<script>alert("文件'.$file.'\n 下载成功！ ");window.close();</script>';
}else {
    echo '<script>alert("文件'.$file.'\n 下载！ ");history.back();</script>';
}
ftp_close($link);                                   //关闭 FTP 连接
}else{                                  //如果下载路径不存在，则向用户展示下载提示框
?>
<script src="js/check.js"></script>
<!--省略部分代码-->
<form id="download" action="#" method="post" onsubmit="return chkdown();">
<input id="down" class="ipt1" type="text" name="savepath" />
<p><span style=" color:#FF0000;">*请填写有效路径（如 C:\Leon\Appserv\)</span></p>
<input class="btn" type="submit" value=" 下载 " />  <input class="btn" type="reset" value="重置" />
</form>
<?php
 }                                                  //结束选择判断
 ?>
```

说明：

（1）远程文件下载操作中，路径输入的 form 表单与远程文件的下载操作均存储于 down.php 文件中。

（2）远程文件下载功能中输入下载路径要按照路径规范书写。如图 14.19 所示，定义的就是一个规范的文件下载路径。



图 14.19　定义正确的远程文件下载路径

14.5.4　文件移动

用户可以选定文件并将其作为对象移动到 FTP 的其他目录中。单击文件列表中的“移动”超链接，进入移动路径页面 movefile.php，从中添加完整的移动路径，然后单击“移动”按钮，将数据提交到 movefile.php，在 movefile.php 中完成文件的移动操作。movefile.php 文件的代码如下：

```
<?php
session_start();                                    //开启全局会话
include_once("config/config.php");                  //加载公共连接文件
```



```
$filename=$_GET["dir"].$_GET["filename"];                  //定义文件路径
$fname=$_GET["filename"];                                  //获取文件名
if(empty($_POST["movepath"])){                             //判断路径是否存在
?>
<form id="movefile" action="#" method="post" onSubmit="return chkpath();">
将<?php echo $filename; ?>移动到：
<p>
   <center>
      <input class="inpt1" id="path" type="text" name="movepath">
   </center>
<p>
   <center>
      <span class="notice">*请填写正确的路径。"如/Appserv/"</span>
   </center>
<p>
   <center>
      <input class="btn2" type="submit" value="移动文件">
   </center>
</form>
<?php }else{
$newpath=$_POST["movepath"].$fname;                        //如果存在则执行移动操作
if(ftp_put($link,$newpath,$filename,FTP_BINARY)){          //将文件移动到指定文件目录
unlink($filename);                                         //删除原来位置的文件
      echo '<script>alert("移动'.$file.'\n 到'.$_POST[movepath].'成功！ ");window.close();</script>';
}else {                                                    //提示移动成功
      echo '<script>alert("移动'.$file.'\n 到'.$_POST[movepath].'失败！ ");history.back();</script>';
}                                                          //提示移动失败
ftp_close($link);                                          //关闭 FTP 连接
 }?>
```

14.5.5　更改文件名称

在移动文件过程中，文件名称冲突可能会导致移动失败，因此还必须提供更改文件名称的功能。单击文件列表中指定文件后的“更改”超链接，跳转到 edit.php 页面中，从中添加新的文件名称，然后单击“更改”按钮，将数据提交到 edit.php，即可完成对文件名称的更改操作。edit.php 文件的关键代码如下：

```
<!--建添加文件名称的表单-->
<table>
<!--省略部分代码-->
<td width="770" align="center"><table   border="0" cellpadding="0" cellspacing="0">
      <tr>
         <td align="center" class="STYLE1">
<form id="form1" name="form1" method="post"
action="edit.php?dir=<?php  echo  $_GET['dir'].'&file='.$_GET['file']  ?>"  onSubmit="return chkamend();">
                  <p>原文件名为：<?php echo $_GET['file']; ?></p>
```



```
                    <p>输入文件名：
                    <input id="news" type="text" name="new" />
                      
                    <input type="submit" name="Submit" value="更改" />
                    </p>
                    <p>注意：要输入文件的全名，包括扩展名</p>
                </form>
            </td>
```

获取 URL 参数，应用 ftp_rename()函数将文件名称更改。其代码如下：

```
<?php
//获取表单提交的文件名称，完成对指定文件名称的更改操作
session_start();                                          //开启全局会话
include_once("config/config.php");                        //加载公共连接文件
$file = $_GET['dir'].$_GET['file'];                       //获取文件完整路径
if($_POST){                                  //判断是否存在以 POST 方式提交的数组
    $new = $_GET['dir'].$_POST['new'];                    //获取新命名文件完整路径
    $result = ftp_rename($link, $file, $new);             //更改名称
    if($result)                                           //判断是否更改成功
    {
        echo '<script>alert(" 文件'.$_GET['file'].'\n 改名为\n'.$_POST['new'].'\n 成功");window.close();</script>';
    }                                                     //提示更改成功
    else{
        echo '<script>alert(" 文件'.$_GET['file'].'\n 改名为\n'.$_POST['new'].'\n 失败");history.back()";</script>';
    }                                                     //提示更改失败
}
?>
```

说明：

更改文件名称时必须添加扩展名，如 index.php，而不是 index。如果更改时未添加扩展名，可能会导致程序无法正常运行。

14.6 技术提炼

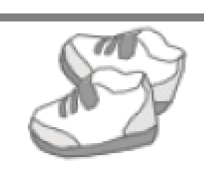
14.6.1 FTP 文件操作技术

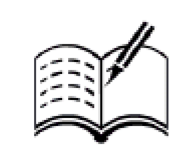
对 FTP 文件的操作，应用的是 PHP 内置的 FTP 文件传输函数库中的函数。下面对其中常用的函数进行详细讲解。

（1）ftp_connect()函数

功能：建立一个新的 FTP 连接，如果成功则返回一个连接表示，否则返回 false。该函数的语法格式如下：

```
resource ftp_connect( string host [, int port [, int timeout]] )
```





Note

其中参数说明如下。

- ☑ host：必要参数，FTP 服务器的地址，后面不应以斜线结尾，前面也不需要用 ftp:// 开头。
- ☑ port：可选参数，FTP 服务器的端口，默认端口为 21。
- ☑ timeout：可选参数，设置网络传输的超时时间限制，默认值为 90 秒，适用于 PHP 4.2.0 及以上版本

例如，应用 ftp_connect()函数改变 FTP 服务器上一个文件的属性，代码如下。

```
<?php
$ftp_server = "127.0.0.1";
$link = ftp_connect($ftp_server) or die("不能连接 FTP 服务器");          //建立 FTP 连接
if($link){
    echo "连接 FTP 服务器成功";
}
ftp_close($link);                                                      //关闭 FTP 连接
?>
```

（2）ftp_login()函数

功能：登录到 FTP 服务器，如果成功返回 true，否则返回 false。该函数的语法格式如下：

```
bool ftp_login( resource ftp_stream, string username, string password )
```

其中参数说明如下。

- ☑ ftp_stream：必要参数，FTP 连接的标识符。
- ☑ username：必要参数，FTP 服务器的用户名。
- ☑ password：必要参数，FTP 服务器的密码。

例如，应用 ftp_login()函数登录到 FTP 服务器，代码如下。

```
<?php
$link = ftp_connect("127.0.0.1");
$login_result = ftp_login($link, "mr", "mrsoft");                      //登录 FTP 服务器
if($login_result) {
    echo "登录成功";
}else {
    echo "登录失败";
}
ftp_close($link);                                                      //关闭 FTP 连接
?>
```

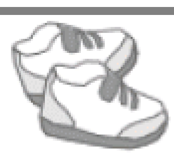
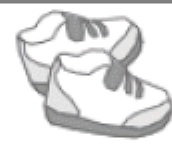
（3）ftp_chdir()函数

功能：在 FTP 服务器上改变当前目录为指定目录，如果成功返回 true，否则返回 false。该函数的语法格式如下：

```
bool ftp_chdir( resource ftp_stream, string directory )
```

应用该函数切换当前目录为指定目录的代码如下：

```
<?php
```

```
$link = ftp_connect("127.0.0.1");                    //连接指定地址
$login_result = ftp_login($link, "mr", "mrsoft");    //连接 FTP 服务器
ftp_chdir($link, "myfile");                          //改变目录
echo "当前目录为".ftp_pwd($link);                     //输出当前的目录名
ftp_close($link);                                    //关闭 FTP 连接
?>
```

Note

（4）ftp_mdtm()函数

功能：检查指定文件的最后修改时间，并以 UNIX 时间戳的方式返回。如果发生错误，或文件不存在则返回-1。某些 FTP 服务器可能会不支持这个特性。该函数的语法格式如下：

```
int ftp_mdtm( resource ftp_stream, string remote_file )
```

其中参数说明如下。

☑ ftp_stream：必要参数，指定 FTP 连接的标识符。

☑ remote_file：必要参数，设置 FTP 服务器的名称。

应用该函数检查文件修改时间的代码如下：

```
<?php
$file = "myfile/mingri.php";                         //设置文件名称
$link = ftp_connect("127.0.0.1");                    //连接指定地址
$login_result = ftp_login($link, "mr", "mrsoft");    //连接 FTP 服务器
$result = ftp_mdtm($link, $file);                    //检查指定文件的最后修改时间
if($result == -1) {                                  //判断查询的文件是否存在
    echo "查询失败";
}else {
    echo $file."的最后修改时间：".date('Y-m-d H:i:s', $result);    //输出最后修改时间
}
ftp_close($link);                                    //关闭 FTP 连接
?>
```

（5）ftp_nlist()函数

功能：返回指定目录下的文件列表，如果成功则返回指定目录下以文件名组成的数组，否则返回 false。该函数的语法格式如下：

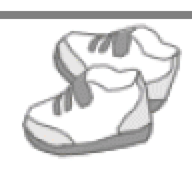
```
array ftp_nlist( resource ftp_stream, string directory )
```

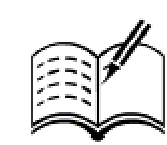
应用 ftp_nlist()函数返回指定目录下的文件列表的代码如下：

```
<?php
$link = ftp_connect("127.0.0.1");                    //连接指定地址
$login_result = ftp_login($link, "mr", "mrsoft");    //连接 FTP 服务器
$result = ftp_nlist($link, "myfile");                //返回指定目录下的文件列表
print_r($result);                                    //输出数组结果集
ftp_close($link);                                    //关闭 FTP 连接
?>
```

（6）ftp_rename()函数

功能：将 FTP 服务器上的一个文件或目录改名，如果成功返回 true，失败则返回 false。该函数的语法格式如下：





Note

```
bool ftp_rename( resource ftp_stream, string oldname, string newname )
```

其中参数说明如下。

☑ ftp_stream：必要参数，FTP 连接的标识符

☑ oldname：必要参数，旧的名称。

☑ newname：必要参数，新的名称。

应用该函数实现文件或目录改名的代码如下：

```
<?php
$link = ftp_connect("127.0.0.1");
$login_result = ftp_login($link, "mr", "mrsoft");
$rename1 = ftp_rename($link, "myfile", "mynewfile");    //将 myfile 文件目录重新命名
if($rename1){                                           //判断目录名称是否被更改
    echo "更改目录名成功";                               //判断文件目录名称是否被更改
}else{
    echo "更改目录名失败";
}
$rename2 = ftp_rename($link, "mr.php", "mingri.php");      //将 mr.php 文件重新命名
if($rename2){                                              //判断文件名称是否被更改
    echo "更改文件名成功";
}else{
    echo "更改文件名失败";
}
ftp_close($link);                                          //关闭 FTP 连接
?>
```

（7）ftp_get()函数

功能：下载 FTP 服务器上的文件，如果成功返回 true，否则返回 false。该函数的语法格式如下：

```
bool ftp_get( resource ftp_stream, string local_file, string remote_file, int mode [, int resumepos] )
```

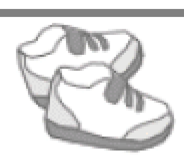
其中参数说明如表 14.2 所示。

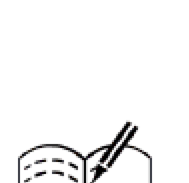
表 14.2　ftp_get()函数的参数说明

参　　数	说　　明
ftp_stream	必要参数，FTP 连接的标识符
local_file	必要参数，保存到本地的文件名
remote_file	必要参数，FTP 服务器上的文件名称
mode	必要参数，传送模式：FTP_ASCII（文本模式）或 FTP_BINARY（二进制模式）
resumepos	可选参数，仅适用于 PHP 4.3.0 以上版本

应用该函数下载指定文件的代码如下：

```
<?php
$down = "mr.php";                                    //设置保存到本地文件名
$file = "myfile/mingri.php";
$link = ftp_connect("127.0.0.1");                    //建立 FTP 连接
```






```
$login_result = ftp_login($link, "mr", "mrsoft");
if(ftp_get($link,$down,$file,FTP_BINARY)) {          //判断是否从 FTP 服务器中下载文件
    echo "下载 $file 成功";
}else {
    echo "下载 $file 失败";
}
ftp_close($link);                                     //关闭 FTP 连接
?>
```

（8）ftp_fput()函数

功能：上传一个已经打开的文件到 FTP 服务器，如果上传成功，返回 true，否则返回 false。该函数的语法格式如下：

```
bool ftp_fput ( resource ftp_stream, string remote_file, resource handle, int mode [, int startpos] )
```

其中参数说明如表 14.3 所示。

表 14.3　ftp_put()函数的参数说明

参　　数	说　　明
ftp_stream	必要参数，ftp 连接的标识符
remote_file	必要参数，本地已经开打文件的句柄
handle	必要参数，ftp 服务器上的文件名称
mode	必要参数，传送模式参数：FTP_ASCII（文本模式）或 FTP_BINARY（二进制模式）
startpos	可选参数，仅适用于 PHP 4.3.0 以上版本

应用 ftp_fput 函数上传指定文件的代码如下：

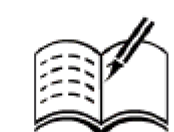
```
<?php
$up = "soft.php";                                    //本地上传文件名
$file = "myfile/mrsoft.php";                         //指定服务器中目录名称
$link = ftp_connect("127.0.0.1");
$login_result = ftp_login($link, "mr", "mrsoft");
if(ftp_put($link, $file, $up,FTP_BINARY)) {          //判断是否上传文件到指定目录中
    echo "上传 $file 成功";
} else {
    echo "上传 $file 失败";
}
ftp_close($link);                                    //关闭 FTP 连接
?>
```

此外，本 FTP 管理系统中还存在其他的 FTP 函数，由于这些函数并非本章重点，所以不再一一介绍。

14.6.2　循环输出技术

所有程序设计中都包含循环控制语句，用于反复多次处理问题。在 FTP 管理系统中，遍历指定文件目录下的内容，应用的是 ftp_nlist()函数，其返回值为数组，所以选择 foreach 语句循环输出文件目录下的内容。foreach 语句的特点就是操作数组，其语法格式如下。






```
foreach (array_expression as $value)
    statement
```

或者

```
foreach (array_expression as $key => $value)
    statement
```



其中，第一种格式遍历给定的 array_expression 数组。每次循环中，当前单元的值被赋给$value 并且数组内部的指针向前移一步（因此下一次循环中将会得到下一个单元）。

第二种格式实现功能与第一种格式相同，只是在输出上略有不同——元素循环输出时，$key 被作为指定元素的键名一同输出。

注意：

foreach 语句不支持“@”来禁止错误信息。

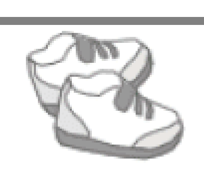
例如，在本系统中应用 foreach 语句来循环输出子目录中文件相关信息的代码如下。

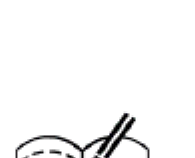
```
 <?php
$dirname = ftp_pwd($link);
if($dirname != '/'){
    $dirname .= '/'; $flist = ftp_rawlist($link, $dirname);
$sdir = array();
foreach($flist as $value){
    $array = explode(' ', rtrim($value));
    $alen = count($array);
    $p = substr($array[0], 0, 1);
    if($p == 'd'){
        if($array[$alen-1] != '.' && $array[$alen-1] != '..'){
            $sdir[] = $array[$alen-1];
        }
    }
}
} foreach($sdir as $value)
    echo'<tr>
         <td height="10" width="30">  </td>
         <td height="25" width="75" align="center" class="STYLE1">
<a href="main.php?dir='.$dirname.$value.'">'.substr($value,0,12).'</a> </td>
         <td width=50><a href="rmkdir.php?dir='.$dirname.$value.'">删除</a></td>
         <td height="5" width="30">  </td>
       </tr>';
?>
```

14.6.3　应用 CSS 设置屏幕滚动条技术

FTP 管理系统的结构和一般连接数据库的 Web 程序不同，其在页面返回的结果依据实际目录中的文件，而一般连接数据库的 Web 程序则是通过连接数据库并应用数据库查询语句返回查询结果；连接数据库的 Web 程序可以通过分页来实现数据的分段显示，

而 FTP 管理系统则很难实现这样的分段显示。面对文件目录中数目众多的目录，FTP





管理系统返回的文件结果将会撑满整个屏幕。如何解决这一问题呢？可以应用 CSS 相关技术设置屏幕的滚动条，来实现文件目录下数据的滚动输出，即当数目大于所给边框长度时，系统的返回结果将会以滚动条的形式输出。

自动判断并设置滚动条的 CSS 的关键代码如下：

```
overflow:auto;                    //设置自动判断并设置滚动条
```

另外也可以设置横向滚动条和纵向滚动条。其语法格式如下：

```
overflow-x:auto;                  //设置横向自动判断并设置滚动条
overflow-y:hidden;                //设置纵向自动判断并隐藏滚动条
```

自动设置滚动条的一般过程如下：

（1）设置 div 布局层，并为该层设置高度。相关代码如下：

```
<div class="content">
<!—HTML 代码部分-->
</div>
```

（2）在<head></head>标签中，添加<style></style>属性标签，并 class 为 content 的 CSS 样式表设置为自动调节高度和宽度。相关代码如下：

```
<head>
 <style   type="text/css">
. content{overflow:auto;height:200px;}
</style>
<head>
```

这样，当类为 content 的 div 容器中的内容高度超过 200px 值时，在 div 的边框会自动显示滚动条，用户可以通过拖拽滚动条来查看类为 content 的 div 中所显示的全部内容。其运行效果如图 14.20 所示。



图 14.20　应用 CSS 设置滚动条

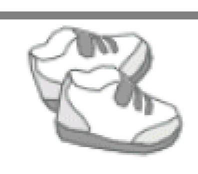


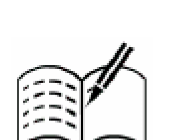
创建 PDF 文档模块

（ 自学视频、源程序：配套资源\mr\15\）

PDF（Portable Document Format）是当前比较流行的电子文档与电子表格的一种标准格式。它提供了完善的压缩处理，无论创建者创建的 PDF 文档使用何种字体、什么样的图片与版式设计，浏览者都可以通过免费的 Adobe Reader 对其进行阅读。本章将要开发的就是一个支持 PDF 文档创建和下载功能的模块。通过本章的学习，读者能够学到以下内容。

- PHP 国际化
- FPDF 的安装与配置
- PDF 文档的创建
- 向 PDF 中插入图片
- 在 PDF 中绘制表格
- 在 PDF 中输出中文
- PDF 文档的下载





15.1 创建 PDF 文档概述

15.1.1 模块概述

创建 PDF 文档模块的重点是将用户添加的文章生成 PDF 文档，并且完成 PDF 文档的下载操作。其中还实现了文章添加、文章浏览等辅助功能，并且实现了网站的国际化（既支持中文操作，也支持英文输出）。

15.1.2 功能结构

创建 PDF 文档模块的功能结构如图 15.1 所示。

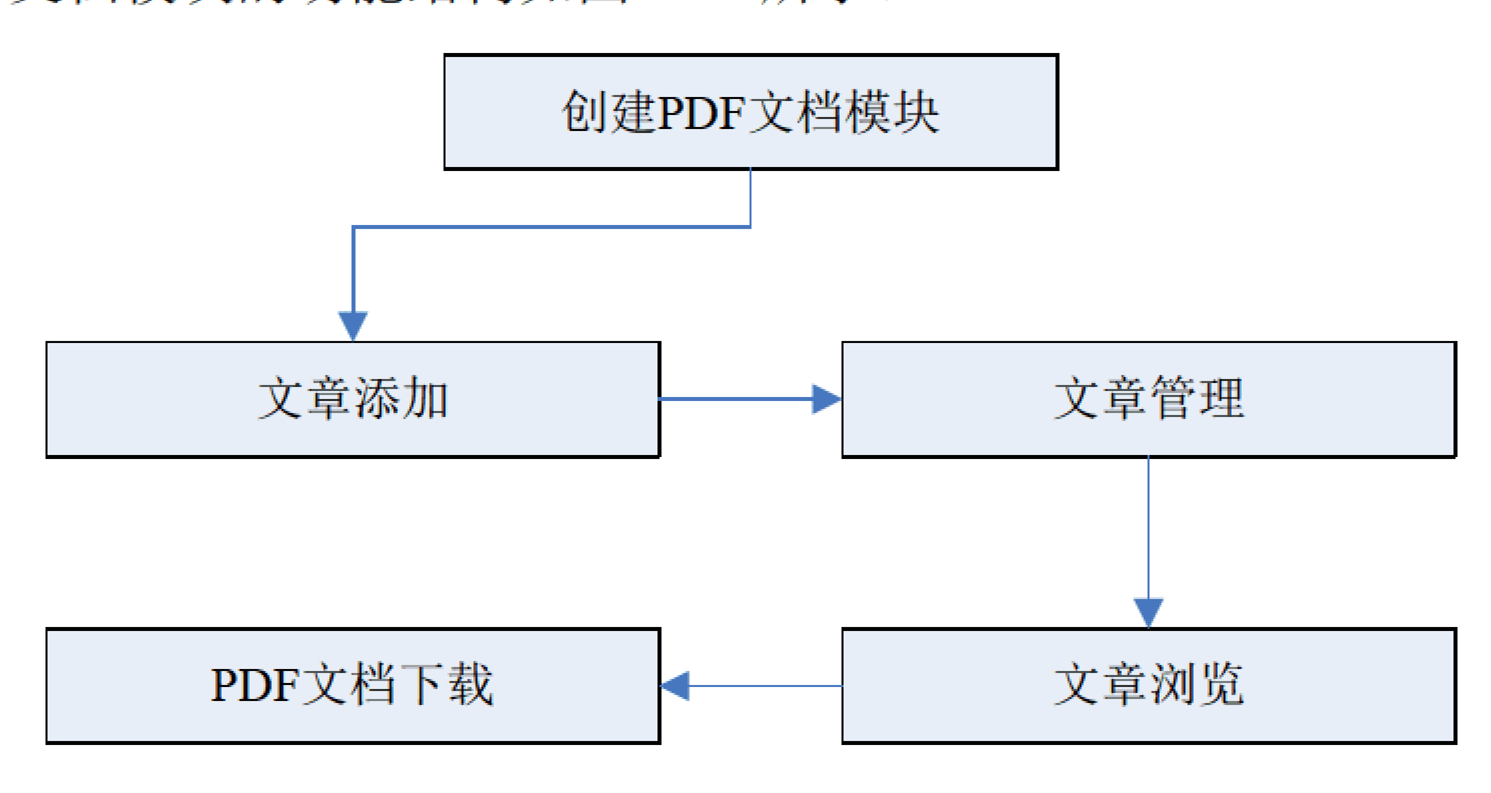


图 15.1 创建 PDF 文档模块功能结构图

15.1.3 程序预览

为了使读者更好地了解创建 PDF 文档模块的功能，下面列出其中一些功能的运行效果图，供广大读者参考。

文章添加页面的运行效果如图 15.2 所示。

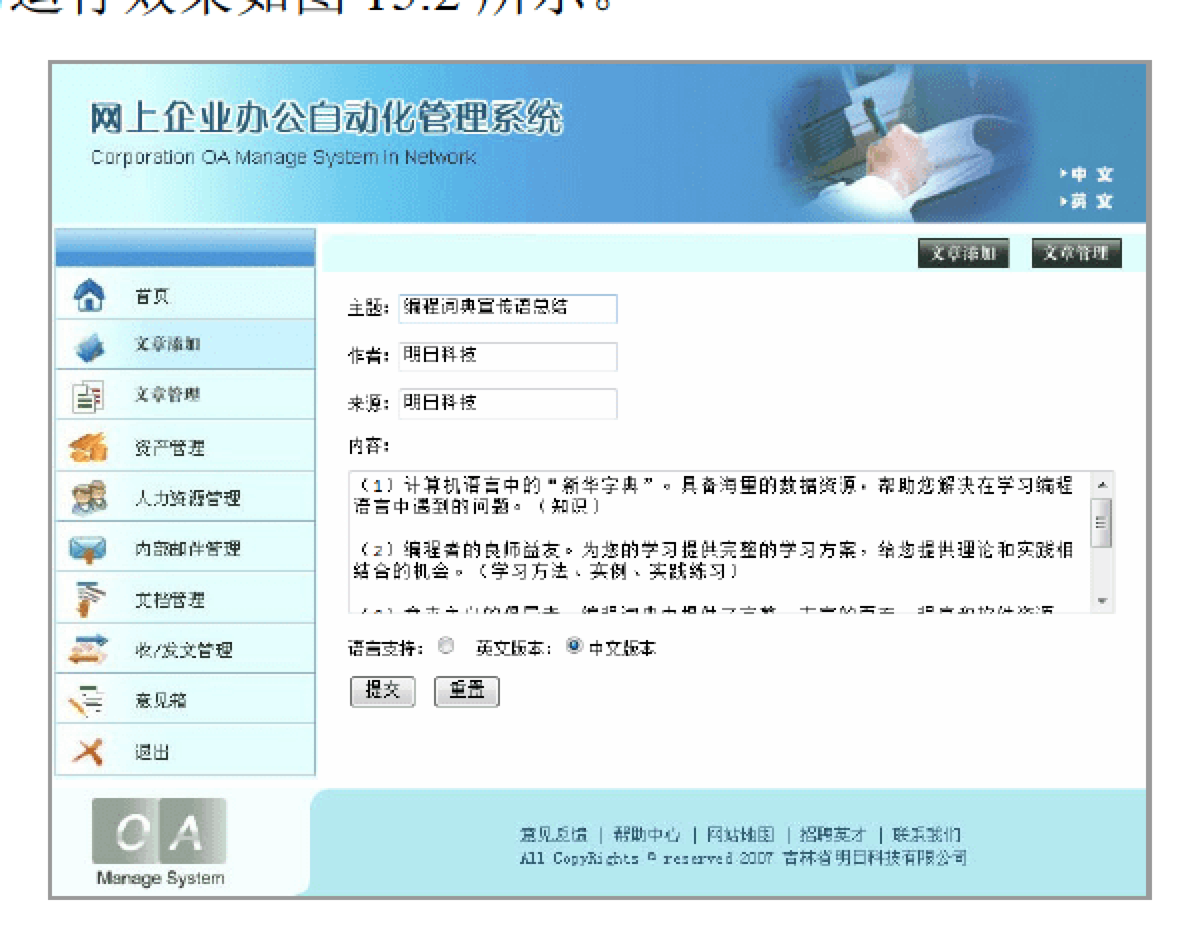
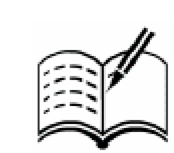


图 15.2 文章添加页面运行效果





文章浏览页面运行效果如图 15.3 所示。

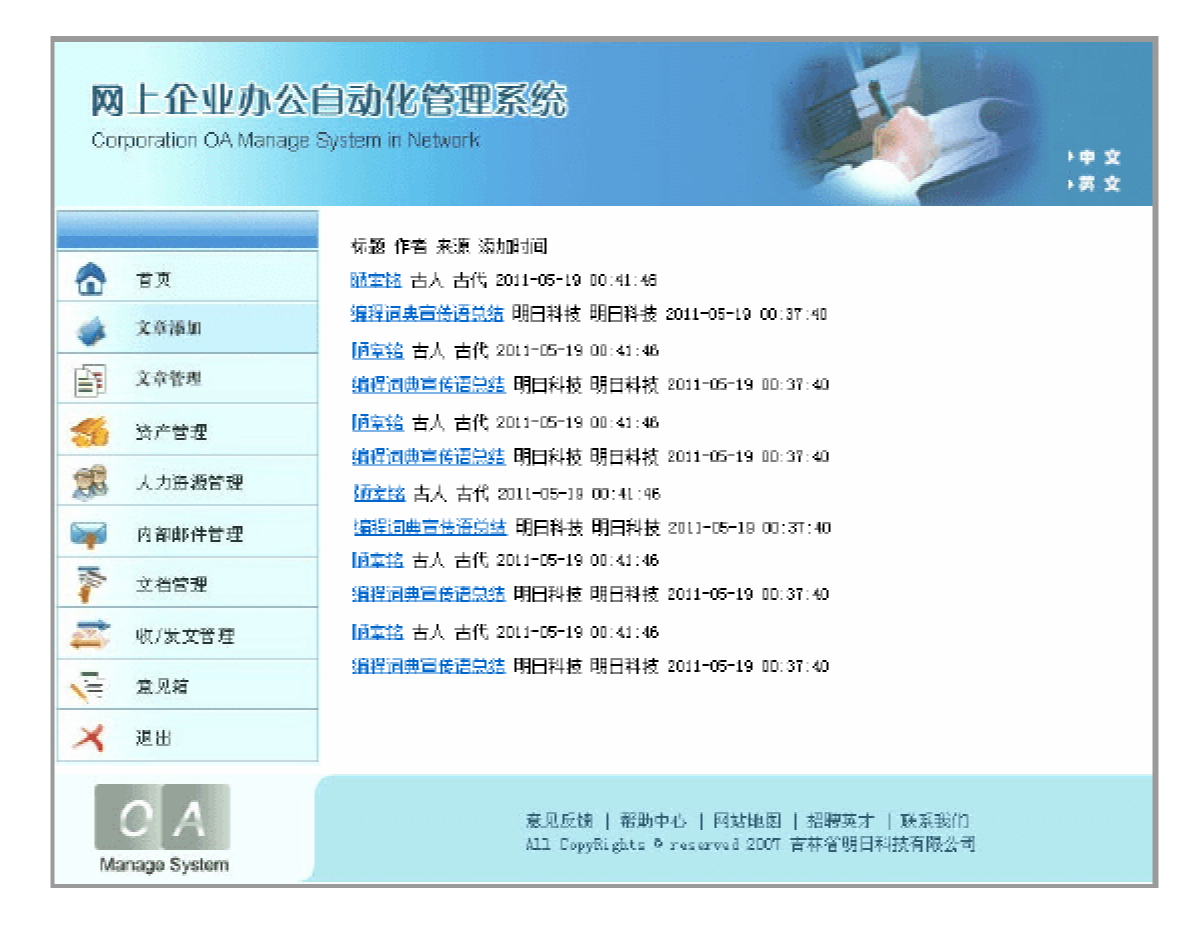


图 15.3　文章浏览页面运行效果

文章内容浏览和 PDF 文档下载的链接如图 15.4 所示。



图 15.4　文章内容浏览和 PDF 文档下载链接

PDF 文档下载的运行效果如图 15.5 所示。

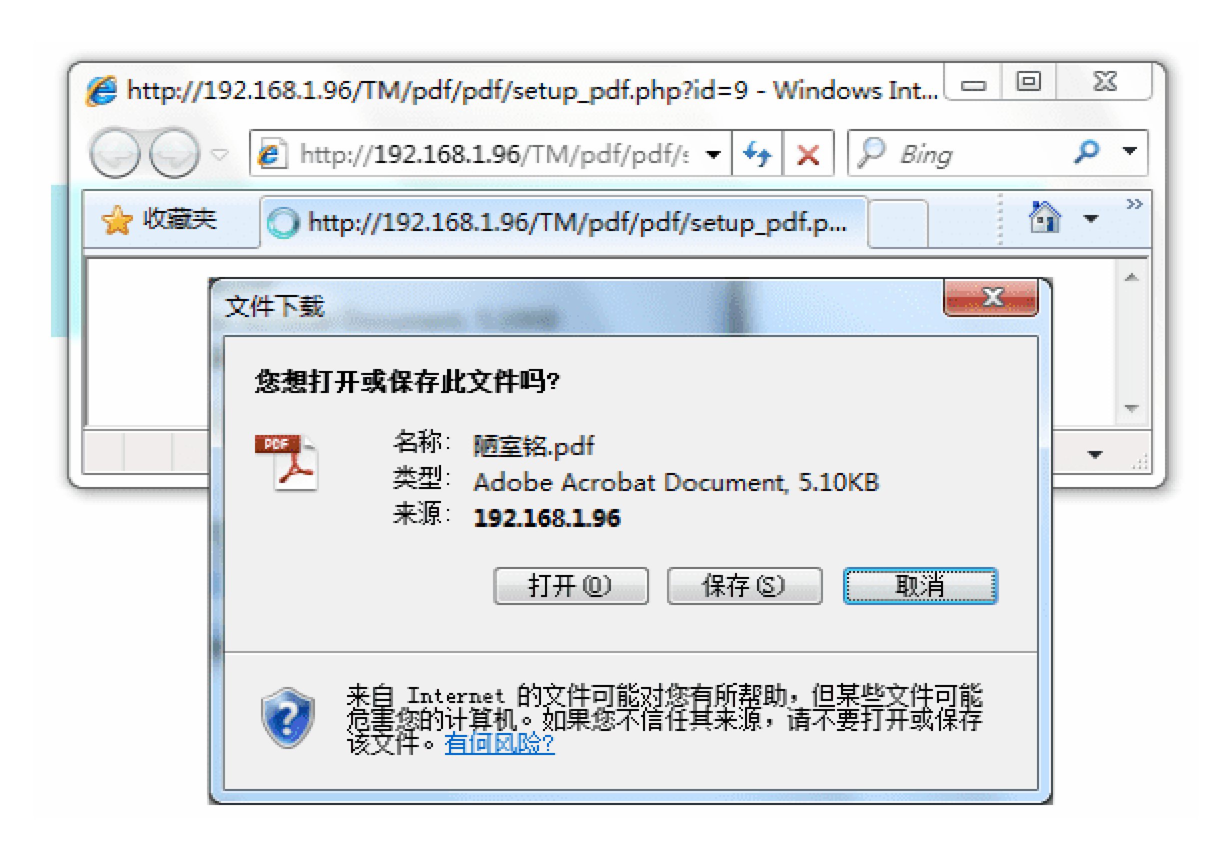


图 15.5　PDF 文档下载的运行效果

通过 Adobe Reader 阅读 PDF 文档的内容，其运行效果如图 15.6 所示。







Note

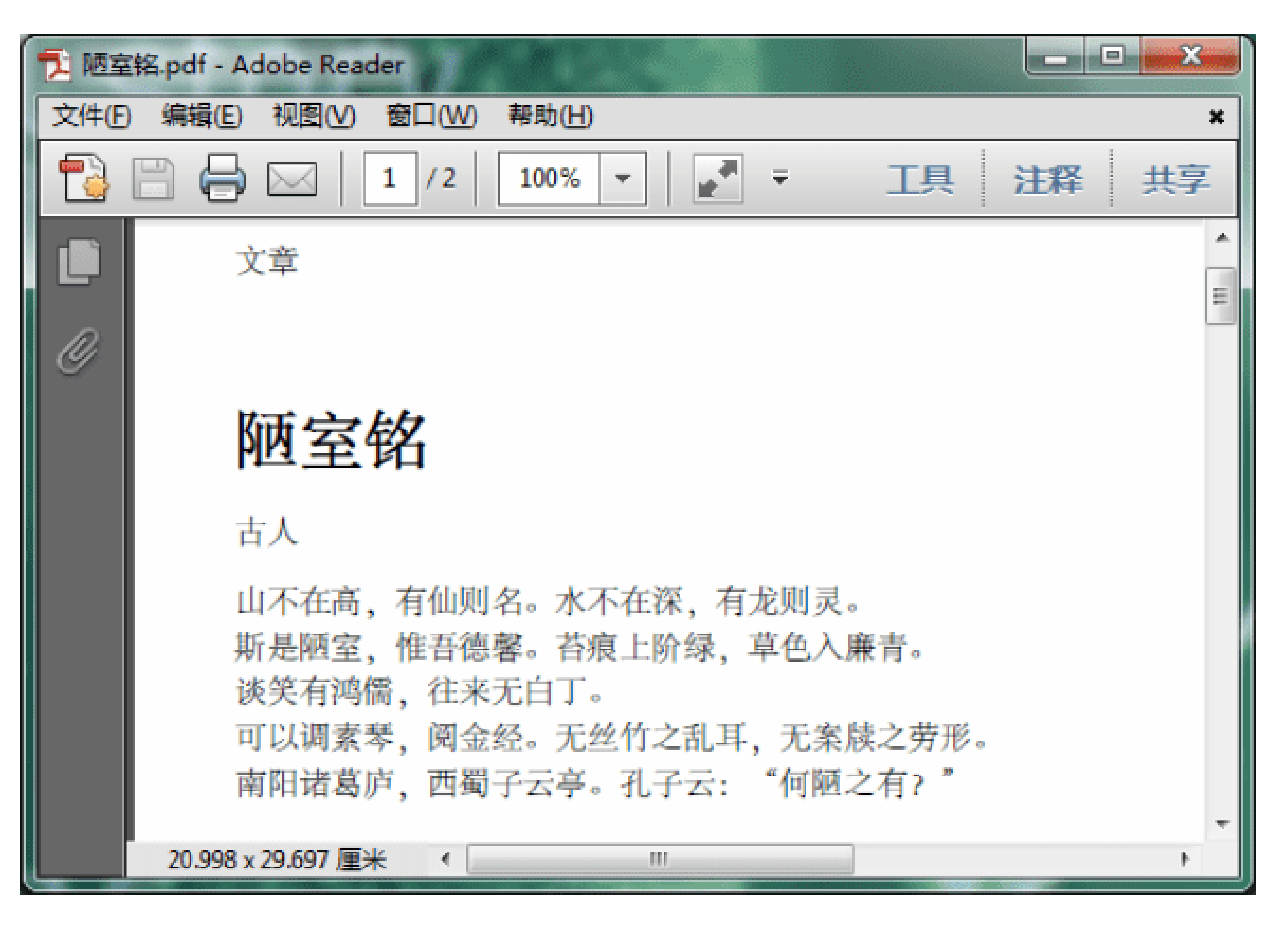


图 15.6　通过 Adobe Reader 浏览 PDF 文档的内容

15.2　数据库设计

在本模块中通过 MySQL 数据库存储用户上传的文章，然后以 PDF 格式将文章内容下载到本地。

15.2.1　创建数据库

在本模块中，创建 db_pdf 数据库存储用户上传的文章数据。db_pdf 数据库中只包含一张数据表：tb_articles。数据库的结构及数据表的情况如图 15.7 所示。



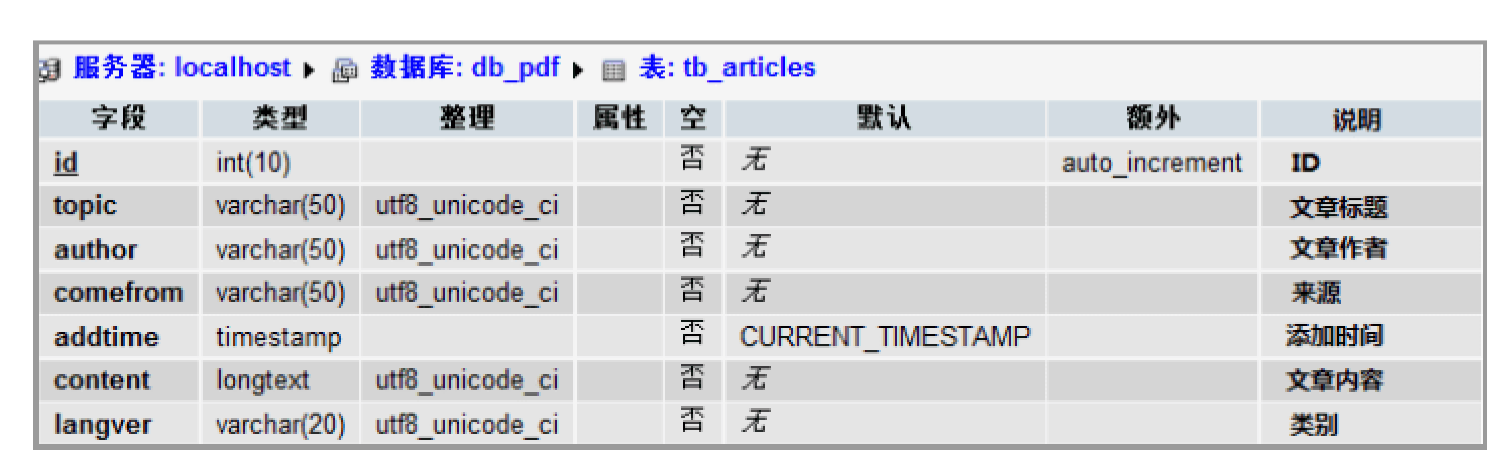
服务器: localhost ▸ 数据库: db_pdf

表	操作	记录数	类型	整理
tb_articles		4	MyISAM	utf8_unicode_ci
1 个表	总计	4	MyISAM	utf8_general_ci

图 15.7　db_pdf 数据库结构

15.2.2　创建数据表

在 tb_articles 数据表中存储用户提交的文章信息，包括 ID、标题、作者、来源、添加时间、内容和类别。其结构如图 15.8 所示。

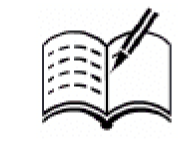


服务器: localhost ▸ 数据库: db_pdf ▸ 表: tb_articles

字段	类型	整理	属性	空	默认	额外	说明
id	int(10)			否	无	auto_increment	ID
topic	varchar(50)	utf8_unicode_ci		否	无		文章标题
author	varchar(50)	utf8_unicode_ci		否	无		文章作者
comefrom	varchar(50)	utf8_unicode_ci		否	无		来源
addtime	timestamp			否	CURRENT_TIMESTAMP		添加时间
content	longtext	utf8_unicode_ci		否	无		文章内容
langver	varchar(20)	utf8_unicode_ci		否	无		类别

图 15.8　tb_articles 数据表的结构





15.2.3 连接数据库

本模块中将连接数据库的操作存储于 connections\conn.php 文件中，设置数据库用户名是 root，密码为 111。conn.php 文件的代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or die('连接失败:' . mysql_error());//连接服务器
mysql_select_db("db_pdf",$conn) or die ('数据库选择失败:' . mysql_error()); //选择数据库
mysql_query("set names utf8");              //设置页面编码格式
?>
```

15.3 模块国际化设置

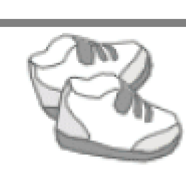
本模块的国际化只是一个辅助功能，目的是让文章的添加和浏览功能支持中文和英文两种语言。本模块国际化设置的步骤如下：

（1）创建 lang 文件夹，编写 lang_ch.php 和 lang_en.php 文件，分别定义中文语言配置和英文语言配置文件，代码参见配套资源中的内容。

（2）创建 config.php 文件，获取浏览器中语言选项的值，根据语言选项的值调用不同的语言配置文件。其代码如下：

```
<?php
session_start();
$lang=$_SERVER['HTTP_ACCEPT_LANGUAGE'];     //获取语言选项
$lang=explode(';',$lang);                   //以“;”对字符串进行分隔
$lang=$lang[0];                             //为数组赋值
$lang=explode(',',$lang);                   //以“,”对字符串进行分隔
$lang=$lang[0];                             //为数组赋值
switch($lang){                          //根据获取到的语言选项的值，为变量赋值
        case "en-us":
                $lang="en";
                break;
        case "zh-cn":
                $lang="ch";
                break;
        default :
                $lang="en";
                break;
}
if(isset($_SESSION['lang'])){
        $lang=$_SESSION['lang'];                //为 SESSION 变量赋值
}
include "lang/lang_$lang.php";              //调用不同的语言配置文件
?>
```

（3）创建 setlang.php 文件，根据超链接中传递的参数，为 SESSION 变量赋值，并且调用不同的操作页面。其代码如下：





Note

```php
<?php
session_start();
switch($_GET['lang']){                    //获取超链接传递的语言选项的值
        case "en":
                $_SESSION['lang']="en";   //根据超链接的值，为 SESSION 变量赋值
                break;
        case "ch":
                $_SESSION['lang']="ch";
                break;
        default :
                $_SESSION['lang']="en";
                break;

}
if($_GET['link']=="addnew"){              //根据超链接的值，跳转到不同操作页面
        header("Location:add_data.php");
}elseif ($_GET['link']=="show"){
        header("Location:look_over.php");
}else{
        header("Location:index.php");
}
?>
```

至此，本模块国际化设置完毕，下面讲解模块具体功能的开发。

15.4 文章浏览

15.4.1 文章浏览概述

文章浏览功能默认输出英文格式的文章标题数据，运行效果如图 15.9 所示。当单击文章标题超链接后，将浏览到文章的详细信息，并且在详细信息页还设置了 PDF 文档下载的超链接，运行效果如图 15.10 所示。

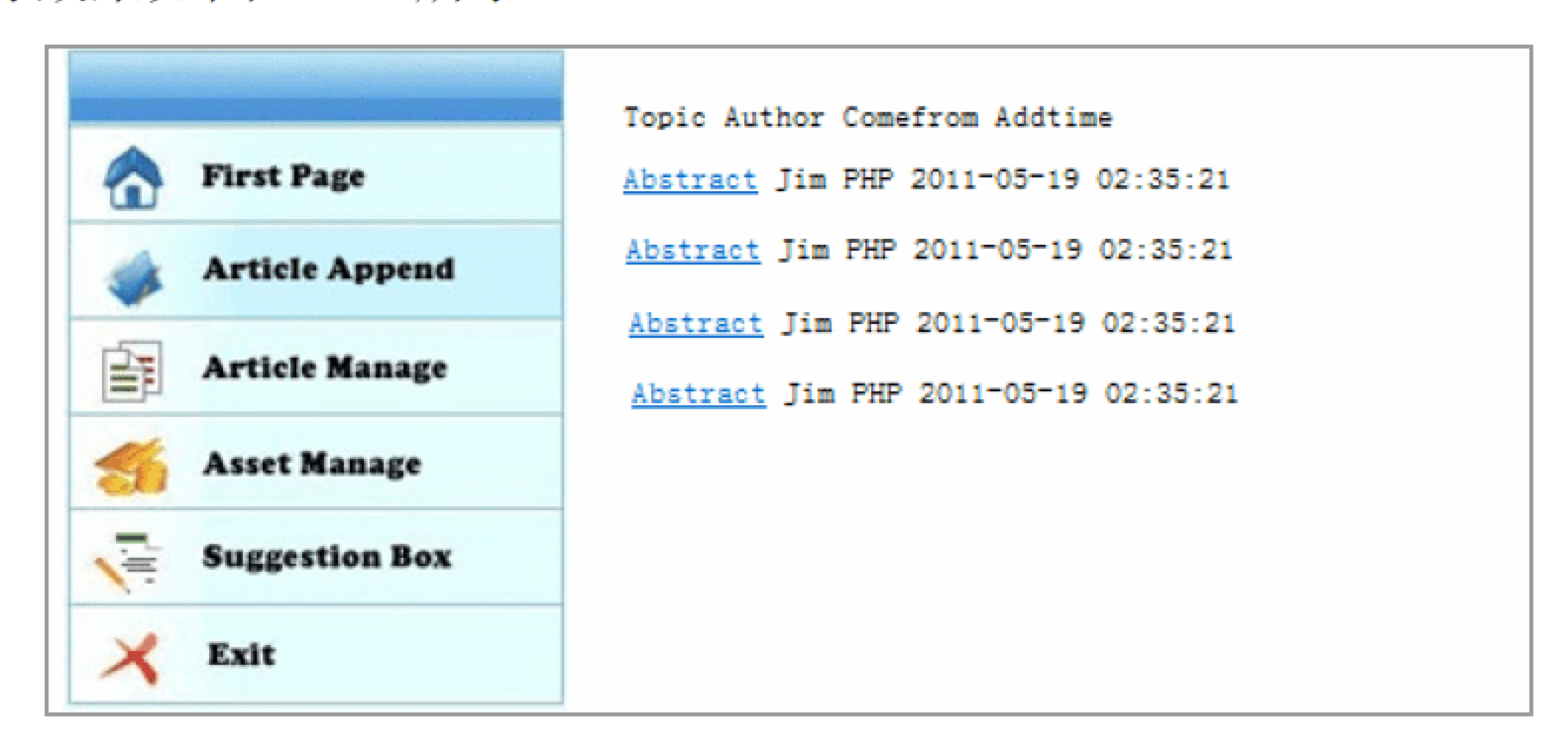
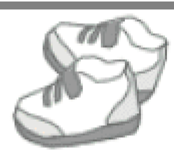


图 15.9 英文格式的文章列表



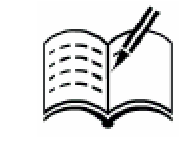


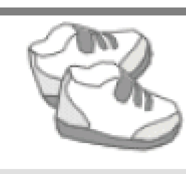


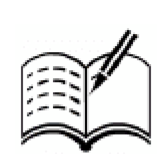
图 15.10　英文格式的文章浏览

15.4.2　文章列表实现过程

创建 index.php 文件，根据选择的语言不同，输出不同的页面和内容。首先，通过 require_once()语句调用不同的包含文件，完成数据库连接和语言配置；然后，执行查询操作，从数据表中读取出指定的数据信息；最后，在页面中输出数据。其完整代码如下：

```
<?php
require_once('Connections/conn.php');                    //包含数据库连接文件
require_once('config.php');                              //包含语言配置文件
$query_rs="select * from tb_articles where langver='$rs_filter' order by id desc";   //定义查询语句
$rs=mysql_query($query_rs,$conn) or die(mysql_error());  //执行查询语句
$row_rs=mysql_fetch_assoc($rs);                          //获取查询结果
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title><?php echo $index_title ?></title>
<link href="css/style.css" rel="stylesheet" type="text/css" />
</head>
<body class="twoColLiqLtHdr">
<div id="container">
<div id="header"><img src="images/<?php if($rs_filter=="en"){echo "OA_021.jpg";}else{echo "OA_02.jpg";}?>"
width="780" height="120" border="0" usemap="#Map" />
<map name="Map" id="Map"><area shape="rect" coords="709,73,776,95" href="setlang.php?lang=ch&link=addnew"/>
<area shape="rect" coords="706,98,772,113" href="setlang.php?lang=en&link=addnew" /></map>
    <!-- end #header --></div>
    <div id="sidebar1">
        <img src="images/<?php if($rs_filter=="en"){echo "OA_061.jpg";}else{echo "OA_06.jpg";}?>"
  width="188" height="396" border="0" usemap="#Map2" />
            <map name="Map2" id="Map2"><area shape="rect" coords="47,110,154,131"
```





Note

```
href="index.php" />
    <area shape="rect" coords="43,71,163,98" href="add_data.php" /></map>
      <!-- end #sidebar1 --></div>
      <div id="mainContent">
         <P><?php echo $index_topic; ?>
                      <?php echo $index_author; ?>
                      <?php echo $index_comefrom; ?>
                      <?php echo $index_addtime; ?>
         </P>
          <?php
         do{
    ?><P>
                   <a   href="look_over.php?id=<?php   echo   $row_rs['id'];?>"><?php   echo
$row_rs['topic'];?></a>
                      <?php echo $row_rs['author'];?>
                      <?php echo $row_rs['comefrom'];?>
                      <?php echo $row_rs['addtime'];?>
         </P>
           <?php
      }while($row_rs=mysql_fetch_assoc($rs));
       mysql_free_result($rs);
       ?>
         <!-- end #mainContent --></div>
      <div id="footer"><img src="images/OA_13-14.jpg" width="780" height="78" />
      <!-- end #footer --></div>
    <!-- end #container --></div>
    </body>
    </html>
```

注意：

index.php 文件中输出的内容，会根据不同语言（中文和英文）传递的参数值进行切换输出。

15.4.3 文章浏览实现过程

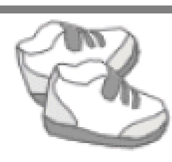
单击文章标题超链接后，将跳转到 look_over.php 文件中，在此文件中根据超链接传递的文章 ID 查询出该文章的详细信息，并输出文章的内容。同时在 look_over.php 文件中创建“下载 PDF 文档”超链接，链接到 setup_pdf.php，在此文件中完成 PDF 文档的生成和下载操作。look_over.php 文件的关键代码如下：

```
<?php
require_once('Connections/conn.php');                    //连接数据库
require_once('config.php');                              //配置语言
$colname_rs='1';                                         //定义变量值
if(isset($_GET['id'])){                                  //检测变量是否被设置
    $colname_rs=(get_magic_quotes_gpc())?$_GET['id']:addslashes($_GET['id']);   //
对数据格式进行转换
```



```
}
$query_rs=sprintf("select * from tb_articles where id=%s",$colname_rs);    //对字符串进行格式化
$rs=mysql_query($query_rs,$conn) or die(mysql_error());              //执行查询操作
$row_rs=mysql_fetch_array($rs);                                      //获取查询结果集
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title><?php echo $index_title ?></title>
<link href="css/style.css" rel="stylesheet" type="text/css" />
</head>
<body class="twoColLiqLtHdr">
<div id="container">
<div id="header"><img src="images/<?php if($rs_filter=="en"){echo "OA_021.jpg";} else{echo "OA_02.jpg";}?>"
width="780" height="120" border="0" usemap="#Map" />
<map name="Map" id="Map"><area shape="rect" coords="708,73,775,95" href="setlang.php?lang=ch&link=addnew" />
<area shape="rect" coords="706,98,772,113" href="setlang.php?lang=en&link=addnew" /></map>
  <!-- end #header --></div>
  <div id="sidebar1">
    <img src="images/<?php if($rs_filter=="en"){echo "OA_061.jpg";}else{echo "OA_06.jpg";}?>"
width="188" height="396" border="0" usemap="#Map2" />
        <map name="Map2" id="Map2"><area shape="rect" coords="47,111,154,132" href="index.php" />
<area shape="rect" coords="43,71,163,98" href="add_data.php" /></map>
  <!-- end #sidebar1 --></div>
  <div id="mainContent">
    <P><?php echo $row_rs['topic'];?></p>
    <P><?php echo $show_author; ?>: <?php echo $row_rs['author'];?></p>
    <P><?php echo $show_comefrom;?>: <?php echo $row_rs['comefrom'];?></p>
    <P><?php
        $len=iconv_strlen($row_rs['content'],'utf-8');     //对文章内容长度进行统计
        if($len>100){
            echo iconv_substr($row_rs['content'],0,100,'utf-8')."......";       // 对 文章内容进行截取输出
        }else{
            echo $row_rs['content'];                    //输出文章内容
        }
    ?></p>
    <P><?php echo $show_upttime;?>: <?php echo $row_rs['addtime'];?></p>
    <P><a href="setup_pdf.php?id=<?php echo $row_rs['id']; ?>"
 target="_blank"><?php echo $show_down;?></a></p>
    <!-- end #mainContent --></div>
  <div id="footer"><img src="images/OA_13-14.jpg" width="780" height="78" />
  <!-- end #footer --></div>
<!-- end #container --></div>
```





```
</body>
</html>
<?php
mysql_free_result($rs);
?>
```



Note

技巧：

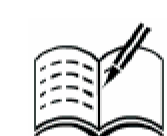
在对文章内容进行统计和截取时，应用的是 iconv_*系列函数。通过 iconv_strlen()函数统计文章的长度，通过 iconv_substr()函数对文章内容进行截取。使用 iconv_*系列函数对字符串进行操作，可以避免截取中文字符串出现乱码的问题。如果这里应用 strlen()函数统计文章长度，通过 substr()函数对文章内容进行截取，那么截取的内容就可能会出现乱码。

15.4.4 PDF 文档的创建和下载

PDF 文档的下载应用的是超链接下载方式，而 PDF 文档的创建在 setup_pdf.php 文件中完成，其根据超链接传递的 ID 值，从数据库中查询出指定文章的详细信息，调用 PDF_Chinese 类中的方法完成 PDF 文档的创建。setup_pdf.php 文件的完整代码如下：

```
<?php
header("Content-Type:text/html;charset=gb2312");          //设置页面编码格式
include('pdf/chinese.php');                               //载入中文组件
include('Connections/conn.php');                          //载入数据库连接文件
class PDF extends PDF_Chinese{                            //继承中文类
        function Header(){                                //定义方法设置标题
                $this->SetFont('gb','',10);               //设置字体
                $this->Write(10,'文章');                  //输出标题内容
                $this->Ln(20);                            //换行
        }
        function Footer(){                                //设置页脚
                $this->SetY(-15);
                $this->SetFont('gb','',10);
                $this->Cell(0,10,'第'.$this->PageNo().'页');
        }
}
$colname_rs_id=$_GET['id'];                               //获取指定数据的 ID
$query_rs_article=sprintf("select * from tb_articles where id=%s",$colname_rs_id);//定义 SQL 语句
$rs_article=mysql_query($query_rs_article,$conn) or die(mysql_error());//执行 SQL 查询语句
$row_rs_article=mysql_fetch_assoc($rs_article);           //获取查询结果
$pdf=new PDF();                                           //实例化 PDF 类
$pdf->AddGBFont();                                        //设置字体
$pdf->Open();                                             //打开文件
$pdf->AliasNbPages();                                     //为每页定义一个别名
$pdf->AddPage();                                          //分页方法
$pdf->SetFont('gb','B',20);                               //设置字体
$pdf->Cell(0,10,iconv("utf-8","gb2312",$row_rs_article['topic']));//输出从数据库中读取的标题数据
$pdf->Ln();                                               //换行
```






```
$pdf->SetFont('gb','',10);
$pdf->Cell(0,10,iconv("utf-8","gb2312",$row_rs_article['author']));
$pdf->Ln();
$pdf->SetFont('gb','',10);
$content= iconv('utf-8','gb2312',$row_rs_article['content']);        //设置编码格式
$pdf->MultiCell(0,5,$content);                                       //输出内容，并且强行终止
$pdf->Ln();
$pdf->AddPage();                                                     //分页
$pdf->Output(iconv("utf-8","gb2312",$row_rs_article['topic']).'.pdf',true);  //生成 PDF 文件
?>
```

说明：

有关通过 FPDF 创建 PDF 文档的详细过程参见 15.6 节。

15.5 文 章 添 加

15.5.1 文章添加概述

文章添加功能将用户提交的文章数据添加到指定的数据表中，其运行效果如图 15.11 所示。

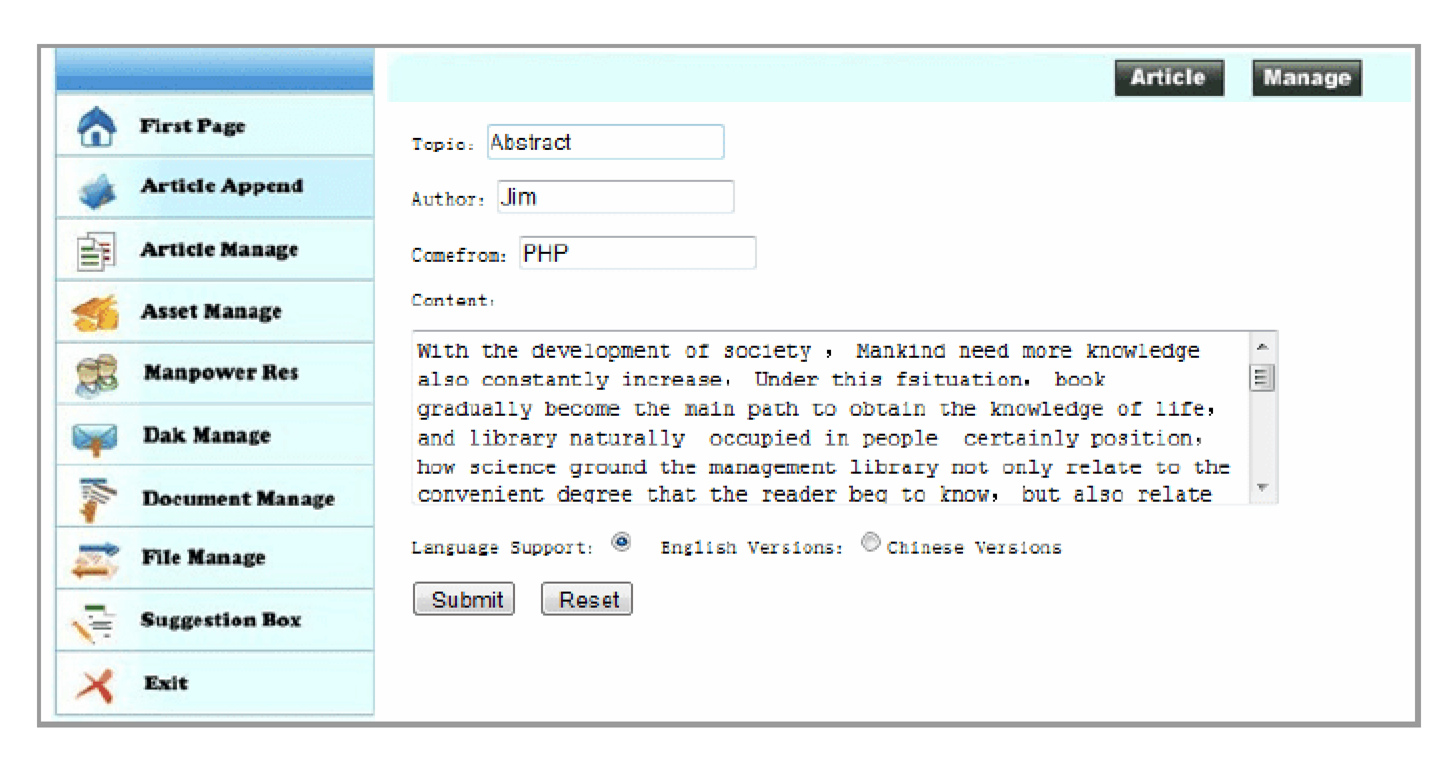


图 15.11 英文格式文章添加的效果

15.5.2 文章添加实现过程

文章添加操作在 add_data.php 文件中完成。可以将其划分为两部分：一是通过表单提交文章数据；二是获取表单提交的数据，将其添加到指定的数据表中。

（1）创建表单的代码如下：

```
<form name="form" method="post" action="">
    <p><?php echo $add_data_topic; ?>: <input type="text" name="topic" value="" /></p>
    <p><?php echo $add_data_author; ?>: <input type="text" name="author" value="" /></p>
    <p><?php echo $add_data_comefrom; ?>: <input type="text" name="comefrom" value="" /></p>
    <p><?php echo $add_data_content; ?>:  </p>
```

Note

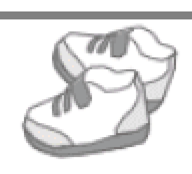
```
            <textarea name="content" id="textarea" cols="65" rows="6"></textarea>
        <p><?php echo $add_data_langsp; ?>:
    <input   type="radio"   name="langver"   value="en"   />  <?php   echo   $add_data_lenver ?>:
    <input type="radio" name="langver" value="ch" /><?php echo $add_data_lchver ?></p>
        <p><input  type="submit"  name="Submit"  value="<?php  echo  $add_data_submit  ?>" />  
    <input type="reset" name="Reset" value="<?php echo $add_data_reset ?>" /></p>
    </form>
```

（2）获取表单提交的数据，将其添加到指定的数据表中。其代码如下：

```
<?php
header("Content-type:text/html;charset=utf-8");      //设置页面编码格式
require_once('Connections/conn.php');                //连接数据库
require_once('config.php');                          //配置语言
//定义格式转换的方法
function GetSQLValueString($theValue,$theType,$theDefinedValue="",$theNotDefinedValue=""){
    $theValue=(!get_magic_quotes_gpc()) ? addslashes($theValue):$theValue;
    switch($theType){
        case "text";
            $theValue=($theValue !="")? "'". $theValue. "'" : "NULL";
            break;
    }
    return $theValue;
}
if(isset($_POST['Submit'])){
    //获取表单中提交的数据，并完成对数据格式的转换
    $insertSQL=sprintf("insert into tb_articles(topic,author,comefrom,content,langver) values (%s,%s,%s,%s,%s)",
        GetSQLValueString($_POST['topic'],"text"),
        GetSQLValueString($_POST['author'],"text"),
        GetSQLValueString($_POST['comefrom'],"text"),
        GetSQLValueString($_POST['content'],"text"),
        GetSQLValueString($_POST['langver'],"text")
        );
        //执行添加语句，将数据添加到数据表中
    $Result1=mysql_query($insertSQL,$conn) or die(mysql_error());
    echo "<script>alert('$add_data_Suc'); window.location.href='add_data.php';</script>";
}
?>
```

说明：

由于本模块中的功能偏少，且重点是讲解 PDF 文档的创建，所以这里只选择文章浏览和文章添加作为辅助内容。如果读者感兴趣，可以在本模块基础上进行二次开发，增加更多、更好的功能。



15.6　技 术 提 炼



在 15.4.4 节中曾讲解了 PDF 文档的创建和下载，但只是从整个模块的角度出发，简单地介绍了 PDF 文档创建和下载所用文件，以及程序代码。下面将对 FPDF 类中的方法进行系统讲解。

15.6.1　FPDF 概述

FPDF 是一个免费的 PDF 操作类，通过它可以完成基本的 PDF 创建功能，并且它还支持中文（需要对相应字体进行配置）。FPDF 类中常用的 PDF 操作函数包括：Open()函数、AddPage()函数，SetFont()函数、Cell()函数和 Output()函数。

可以通过访问 http://www.fpdf.org/来下载 FPDF，然后将下载的文件直接解压到某一文件夹即可。

15.6.2　创建简单的 PDF 文档

通过 FPDF 创建 PDF 文档，首先要实例化 FPDF 类，然后调用类中的函数（Open()函数、AddPage()函数、SetFont()函数、Cell()函数和 Output()函数）完成文档的创建。

（1）创建 FPDF 对象的语法格式如下：

```
$pdf = new FPDF([string page-orientation [, string measure-unit [, string page-format]]])
```

其中参数说明如表 15.1 所示。

表 15.1　创建 FPDF 对象的参数说明

参　　数	说　　明
page-orientation	设置PDF文档的排版。可选值：P，表示竖向；L，表示横向
measure-unit	设置文档中位置的计量单位。可选值：pt，点；mm，毫米；cm，厘米；in，英寸
page-format	设置PDF文档纸张的类型。可选值：表示纸张类型的字符串，例如A4、A5、Letter；一个包含有两个元素的二维数组，直接指明纸张的大小

（2）Open()函数用于标示开始创建 PDF 文档。其语法格式如下：

```
$pdf->Open()
```

（3）AddPage()函数用于为 PDF 文档添加一个新页。其语法格式如下：

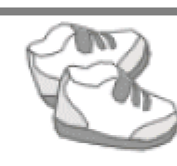
```
$pdf ->AddPage([string page-orientation])
```

其中，page-orientation 表示创建的 PDF 文档是横向还是竖向的，与上述创建对象时的使用方法相同。

（4）SetFont()函数用于设置当前使用的字体。其语法格式如下：

```
$pdf->SetFont(string font [, string style [, float size]]);
```

参数说明如下。




- ☑ font：字体。
- ☑ style：样式。其可选值如下；如果没有指定 style 的值，则默认为普通的形式。
  - ✧ B：粗体。
  - ✧ I：斜体。
  - ✧ U：下滑线。
- ☑ size：字体的大小。如果不指定，则取默认值 12pt。

（5）Cell()函数用于为当前 PDF 文档增加一个单元格。其语法格式如下：

```
$pdf->Cell(float w [, float h [, string txt [, mixed border [, int ln [, string align [, int fill [, mixed link]]]]]]])
```

其中参数说明如表 15.2 所示。

表 15.2　Cell()函数的参数说明

参　数	说　明
w	表示单元格宽度
h	表示单元格高度
txt	表示单元格中添加的文本字符串
bolder	表示单元格的边框。设置为 0 表示无边框，设置为 1 表示边框；或者 L 表示左边框，T 表示顶部边框，R 表示右边框，B 表示底部边框
ln	表示换行后，设置目前的位置。可选值为：0 表示向右移动；1 表示定位到下一行的开始；2 表示向下移动。默认值为 0
align	设置字符串的对齐方式。可选值为：L 表示左对齐；C 表示居中；R 表示右对齐。默认值为 L
fill	设置单元格的背景。可选值为：0 表示透明；1 表示不透明。默认值为 0
link	设置一个 URL 链接或者由 AddLink 函数添加的内部链接

（6）Output()函数用于输出 PDF 文档。其语法格式如下：

```
$pdf->Output([string filename [, bool download]])
```

参数说明如下。

- ☑ filename：要存储的文件名。如果不指定文件名，则浏览时会在 IE 中直接打开。
- ☑ download：指定是否一定要用户下载查看，默认为 false。

下面应用上述介绍的函数，创建一个 PDF 文档，输出一段英文字符串。example1.php 的代码如下：

```
<?php
define('FPDF_FONTPATH','font/');          //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');             //包含 fpdf 类库文件
$pdf=new FPDF('P', 'mm', 'A4');           //创建新的 FPDF 对象，竖向放纸，单位为 mm，纸
张大小 A4
$pdf->Open();                             //开始创建 PDF
$pdf->AddPage();                          //增加一页
$pdf->SetFont('Courier','I',20);          //设置字体样式
$pdf->Cell(0,0,'mrsoft!');                //增加一个单元格
$pdf->Output();                           //输出 PDF 到浏览器
?>
```

运行结果如图 15.12 所示。





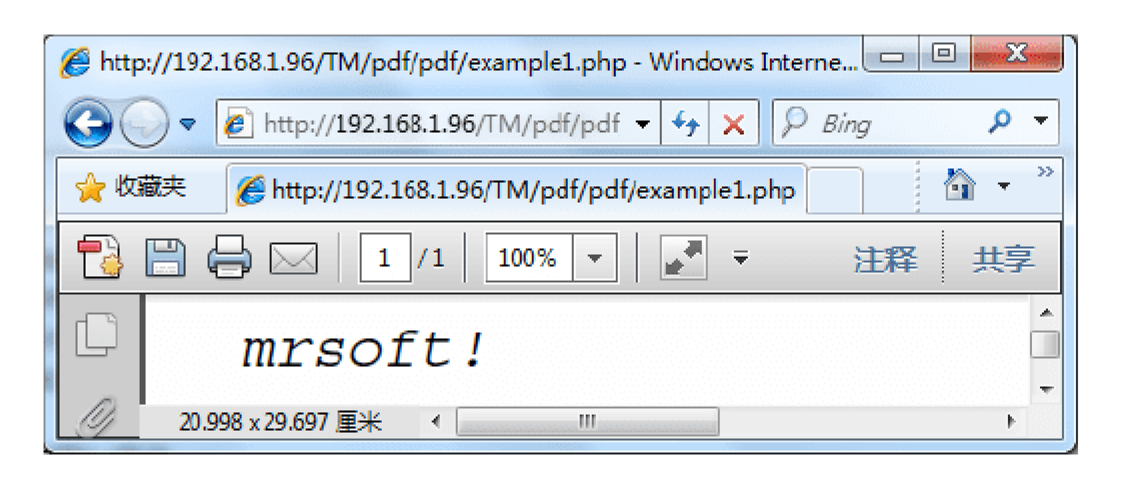


图 15.12　简单的 PDF 文档

技巧：

上面的示例是将 PDF 文档的内容在 IE 浏览器中输出。如果将上述代码中的"$pdf->Output();"修改为"$pdf->Output('mrsoft.pdf',true);"，然后定义 PDF 文档名称为 mrsoft.pdf，设置用户下载查看，即可将 PDF 文档保存到服务器并提示用户下载。

15.6.3　在 PDF 中插入图片

在 FPDF 中通过 Image()函数向 PDF 中插入图片。其语法格式如下：

```
Image(string file, float x, float y, float width, float height)
```

参数说明如下。

☑ x 和 y：图片所在位置的坐标。

☑ width 和 height：图片的宽度和高度。

下面应用 Image()函数向 PDF 文档中插入一个图片，并且将 PDF 在 IE 浏览器中直接输出。example2.php 的代码如下：

```
<?php
define('FPDF_FONTPATH','font/');                //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');                   //包含 fpdf 类库文件
$pdf=new FPDF('P', 'mm', 'A4'); //创建新的 FPDF 对象，竖向放纸，单位为 mm，纸张大小 A4
$pdf->Open();                                   //创建 PDF
$pdf->AddPage();                                //增加一页
$pdf->SetFont('Courier','I',20);                //设置字体样式
$pdf->Image('images/as.jpg',10,10,80,80);       //插入图片
$pdf->Output();                                 //输出 PDF  到浏览器
?>
```

运行结果如图 15.13 所示。

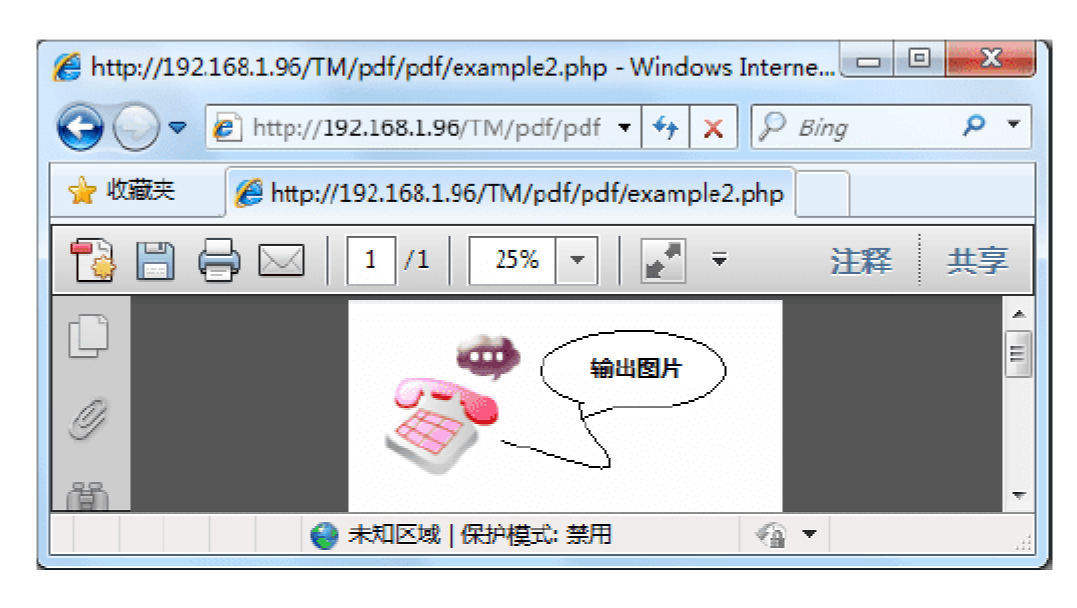
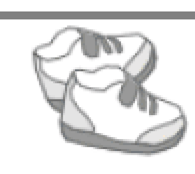
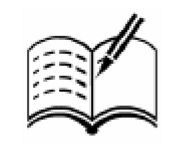


图 15.13　在 PDF 中插入图片





Note

15.6.4 设置 PDF 文档的页眉与页脚

在 FPDF 中，页眉与页脚是通过重写 FPDF 类中的 Header()方法和 Footer()方法来实现的。因为 FPDF 类中的 Header()方法和 Footer()方法虽然存在，但是方法体中没有任何内容，所以设置 PDF 文档的页眉与页脚需要重写 FPDF 类中的 Header()方法和 Footer()方法，完成页眉与页脚的设置。

下面创建 PDF 类，继承 FPDF 类，在 PDF 类中对 Header()方法和 Footer()方法进行重写，完成页眉与页脚的设置。然后，创建 PDF 文档，输出一行英文字符串和一个图片。example3.php 的代码如下：

```
<?php
define('FPDF_FONTPATH','font/');                 //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');                    //包含 fpdf 类库文件
class PDF extends FPDF{                          //继承 FPDF 类，重写页眉与页脚的方法
    function Header(){                           //设置页眉
        $this->SetFont('Arial','B',15);          //设置页眉字体
        $this->Write(10,'Title');                //写入页眉文字
        $this->Ln(20);                           //换行
    }
    function Footer(){                           //设置页脚
        $this->SetY(-15);                        //设置页脚所在位置
        $this->SetFont('Arial','I',8);           //设置页脚字体
        $this->Cell(0,10,'Page - '.$this->PageNo()); //输出当前页码作为页脚内容
    }
}
$pdf=new PDF('P', 'mm', 'A4'); //创建新的 FPDF 对象，竖向放纸，单位为 mm，纸张大小 A4
$pdf->Open();                                    //创建 PDF
$pdf->AddPage();                                 //增加一页
$pdf->SetFont('Courier','I',20);                 //设置字体样式
$pdf->Cell(30,30,'MRSOFT!');                     //输出字符串
$pdf->Image('images/as.jpg',30,50,80,80);        //插入图片
$pdf->Output(); //输出 PDF 到浏览器
?>
```

运行结果如图 15.14 所示。

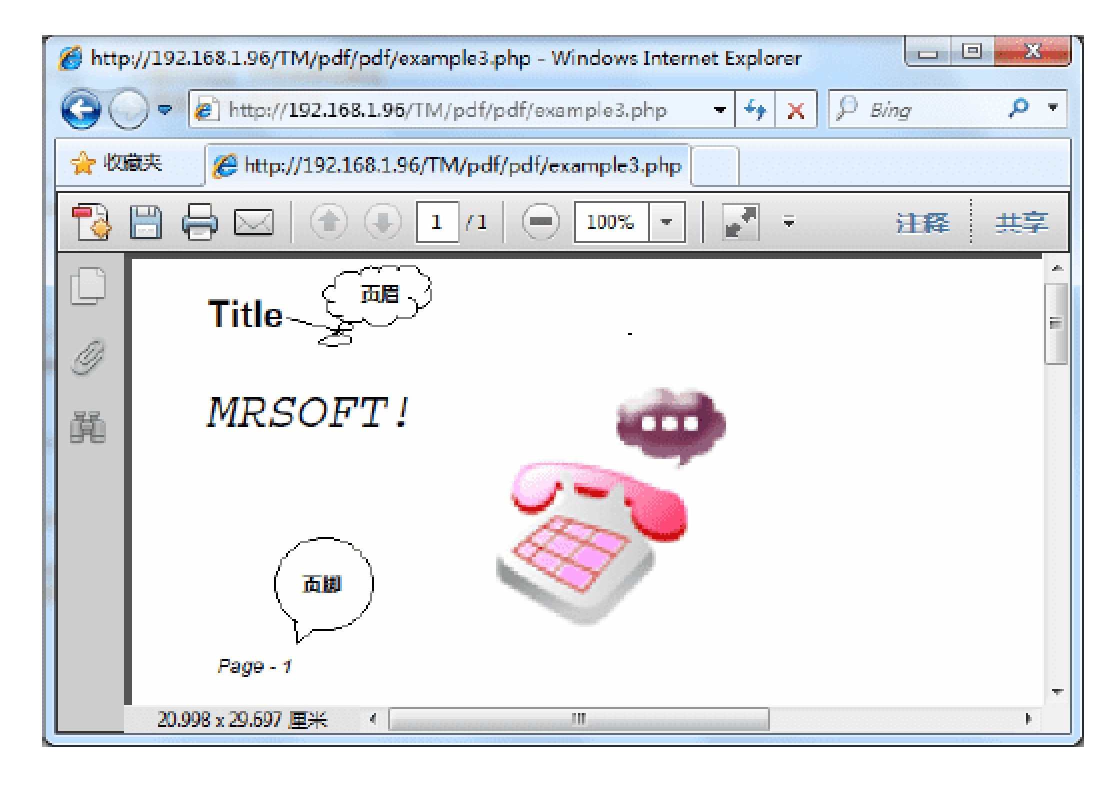
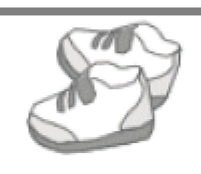


图 15.14　设置 PDF 的页眉和页脚





在设置 PDF 的页眉与页脚时，引入了 Write()、Ln()、PageNo()和 SetY()4 个方法。

（1）Write()方法

Write()方法用于输出字符串，当到达文档的右边位置或者遇到（\n）时，会自动换行。其语法格式如下：

```
Write(float h, string txt [, mixed link])
```

参数说明如下。

- ☑ h：定义字符串的行列高度。
- ☑ txt：指定输出的字符串。
- ☑ link：设置链接的网页或者 AddLink()方法的标识符。

（2）Ln()方法

Ln()方法用于换行操作。其语法格式如下：

```
Ln([float h])
```

其中，参数 h 用于设置行的高度，默认值为最后输出行的高度。

（3）PageNo()方法

PageNo()方法用于返回当前页码。其语法格式如下：

```
int PageNo();
```

（4）SetY()方法

SetY()方法用于设置页脚的坐标。其语法格式如下：

```
SetY(float y);
```

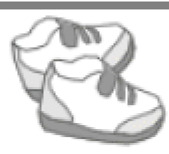
其中，参数 y 为页面上的 Y 坐标，单位为 mm。如果 y 为负数，则表示从页面底部向上的距离。例如，上面的示例中设置 y 的值为-15，即页脚所在位置距页面底部 15mm。

15.6.5 在 PDF 文档中绘制表格

通过 FPDF 绘制表格的方法与输出文字的方法类似，在输出文字时往往指定 Cell()方法的边框参数为 0，即不输出边框；在绘制表格时往往将其设置成大于 0 的整数，用于表示边框，这就是 PDF 文档中表格的单元格。把这些单元格组合到一起，就构成了表格。

下面看一个输出表格的示例 example4.php，其代码如下：

```
<?php
define('FPDF_FONTPATH','font/');                    //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');                       //包含 fpdf 类库文件
class PDF extends FPDF{
    function Header(){                              //设置页眉
        $this->SetFont('Arial','B',15);             //设置页眉字体
        $this->Write(10,'Pay Slip');                //写入页眉文字
        $this->Ln(20);                              //换行
    }
    function Footer(){                              //设置页脚
```




Note

```
            $this->SetY(-15);                          //设置页脚所在位置
            $this->SetFont('Arial','I',8);             //设置页脚字体
            $this->Cell(0,10,'Page - '.$this->PageNo()); //输出当前页码作为页脚内容
        }
    }
    $pdf=new PDF('P', 'mm', 'A4'); //创建新的 FPDF 对象，竖向放纸，单位为 mm，纸张大小 A4
    $pdf->Open();                                      //开始创建 PDF
    $pdf->AddPage();                                   //增加一页
    $pdf->SetFont('Arial','',14);                      //设置字体样式
    $header=array('Name','Age','Duty','Laborage');     //设置表头
    $data=array();                                     //设置表体
    $data[0] = array('Simon','24','employee','5,000.00');
    $data[1] = array('Elaine','25','Manager','6,000.00');
    $data[2] = array('Susan','25','employee','4,000.00');
    $data[3] = array('David','26','Manager','8,000.00');
    $width=array(40,40,40,40);                         //设置每列宽度
    for($i=0;$i<count($header);$i++)                   //循环输出表头
    $pdf->Cell($width[$i],6,$header[$i],1);
    $pdf->Ln();
    foreach($data as $row){                            //循环输出表体
        $pdf->Cell($width[0],6,$row[0],1);
        $pdf->Cell($width[1],6,$row[1],1);
        $pdf->Cell($width[2],6,$row[2],1);
        $pdf->Cell($width[3],6,$row[3],1);
        $pdf->Ln();
    }
    $pdf->Output();                                    //输出 PDF 到浏览器
    ?>
```

运行结果如图 15.15 所示。

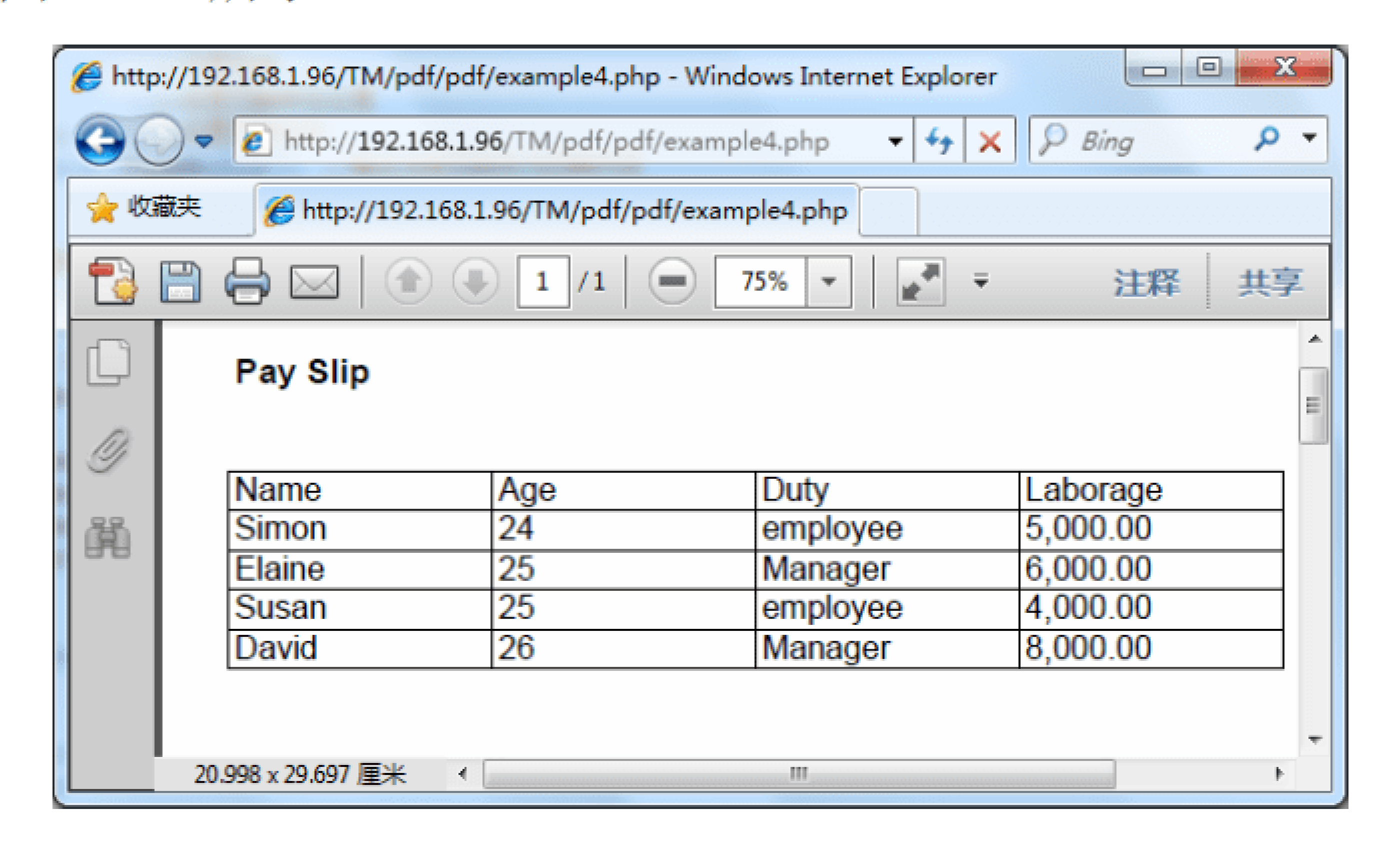


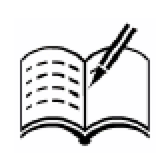
图 15.15　在 PDF 中绘制表格

15.6.6　在 PDF 文档中输出中文

如果应用前面讲解的技术，直接通过 FPDF 在 PDF 文档中输出中文，那么输出的中文字符串将是乱码。解决此问题的方法是应用 FPDF 提供的一个中文插件，继承插件中的 PDF_Chinese 类，在 PDF 文档中输出中文。



首先下载并解压此插件，复制其中的 chinese.php 文件，将其存储于项目的指定文件夹下；然后通过 include 语句在创建 PDF 文档的文件中载入 chinese.php；最后创建 PDF 类，继承 PDF_Chinese 类，在 PDF 文档中输出中文。示例 example5.php 的代码如下：



Note

```
<?php
include('pdf/chinese.php');                      //载入中文插件
class PDF extends PDF_Chinese{                   //继承中文插件类
        function Header(){                       //设置页眉
                $this->SetFont('GB','',10);      //设置字体
                $this->Write(10,'FPDF  中文测试');  //设置页眉的内容
                $this->Ln(20);                   //执行换行操作
        }
        function Footer(){                       //设定页脚
                $this->SetY(-15);                //设置页脚的输出坐标
                $this->SetFont('GB','',10);      //设置字体
                $this->Cell(0,10,'第'.$this->PageNo().'页'); //设置页脚输出的内容
        }
}
$pdf=new PDF();                                  //创建 PDF  文档
$pdf->AddGBFont();                               //添加字体
$pdf->Open();                                    //开启文档
$pdf->AliasNbPages();                            //为每个页面定义一个别名
$pdf->AddPage();                                 //添加页
$pdf->SetFont('GB','I',20);                      //设置字体
$str='山不在高，有仙则名。水不在深，有龙则灵。斯是陋室，惟吾德馨。苔痕上阶绿，草色入帘青。谈笑有鸿儒，往来无白丁。可以调素琴，阅金经。无丝竹之乱耳，无案牍之劳形。南阳诸葛庐，西蜀子云亭。孔子云：“何陋之有？”';
$pdf->Write(10,$str);                            //输出中文
$pdf->Output();
?>
```

运行结果如图 15.16 所示。

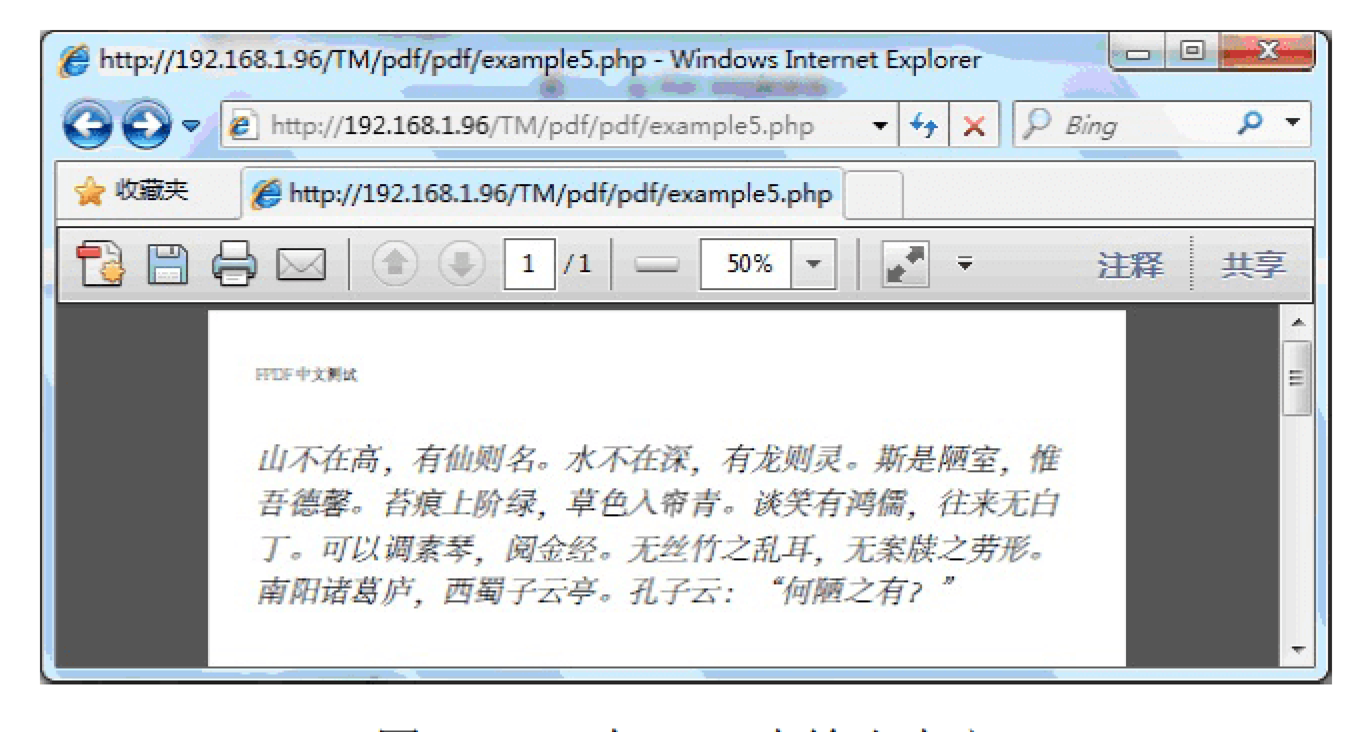
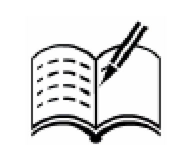


图 15.16　在 PDF 中输出中文

注意：

在创建生成 PDF 文档的文件（example5.php）时，页面的编码格式要设置成 gb2312；如果页面的编码格式是 utf-8，那么即使继承了 PDF_Chinese 类，输出的中文仍旧是乱码。





Note

15.6.7　PDF 文档下载

要实现 PDF 文档的下载，只需在通过 Output()函数输出 PDF 文档时添加两个参数即可。第一个参数定义下载的 PDF 文档的名称；第二个参数设置为 ture，提醒用户下载查看，如果将第二个参数设置为 false，那么生成的 PDF 文档会直接存储于服务器中，不会提醒用户下载查看。

下面对示例 example5.php 进行修改，将其生成的 PDF 文档命名为“程序测试.pdf”，并且提醒用户下载查看。修改的关键代码如下：

```
<?php
include('pdf/chinese.php');                    //载入中文插件
//省略了部分代码
$str='山不在高，有仙则名。水不在深，有龙则灵。斯是陋室，惟吾德馨。苔痕上阶绿，草色入帘青。谈笑有鸿儒，往来无白丁。可以调素琴，阅金经。无丝竹之乱耳，无案牍之劳形。南阳诸葛庐，西蜀子云亭。孔子云：“何陋之有？”  ';
$pdf->Write(10,$str);                          //输出中文
$pdf->Output('程序测试.pdf',true);
?>
```

修改后的示例存储于 example6.php 文件中，运行此示例将输出如图 15.17 所示的 PDF 文件下载页面。

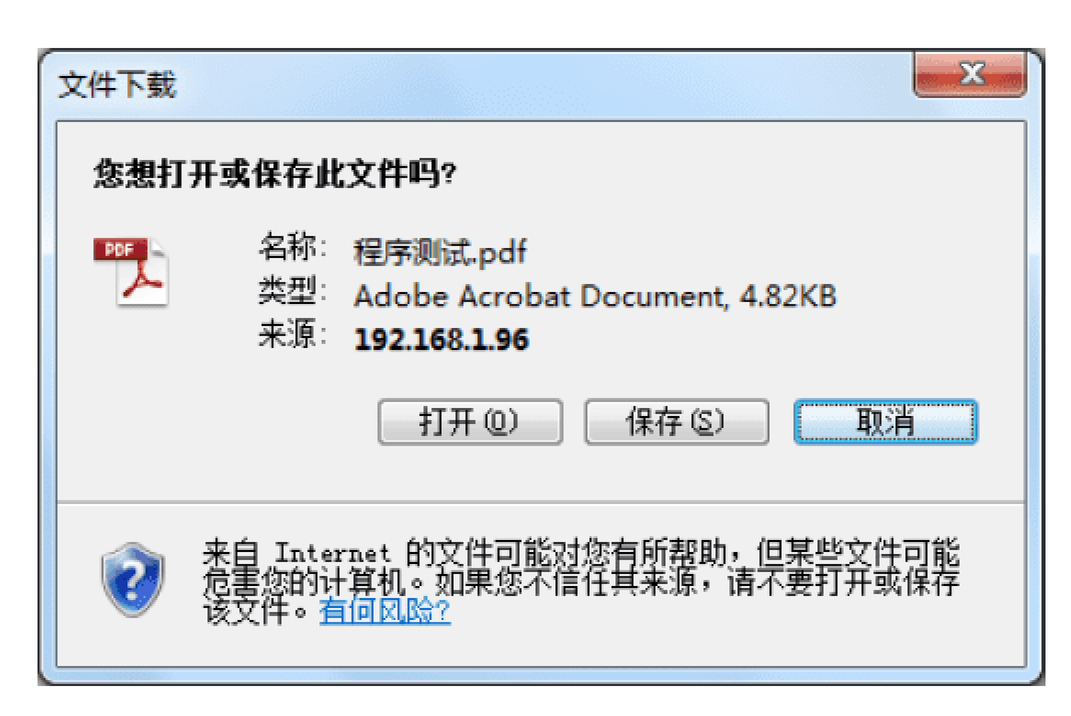


图 15.17　PDF 文档下载